

Adilson Sakahi Ohfugi

Garras Articuladas para Robôs Manipuladores : Análise Cinemática e de
Forças, Sensoreamento e Controle de Posição e Esforços na
Preensão de Objetos

Dissertação apresentada à Faculdade de
Engenharia Elétrica da Universidade
Estadual de Campinas como parte dos
requisitos exigidos para obtenção do título
de MESTRE EM ENGENHARIA ELÉTRICA.

Este exemplar corresponde à redação final de:

defendida

Adilson Sakahi Ohfugi

aprovada pela Co.

Julgada:

20.08.91.

Orientador

Orientador: Prof. Dr. Álvaro Geraldo Badan Palhares

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA
DEPARTAMENTO DE SISTEMAS E CONTROLE DE ENERGIA

Campinas - SP
Agosto 1991

UNICAMP
BIBLIOTECA CENTRAL

Dedico este trabalho, com carinho, para:

Meus Pais:

Motomo e Ramaco, que me apoiaram sempre, com carinho, amizade e dedicação.

Meus Irmãos:

Edna, Elizabete e Roberto, pela amizade e apoio aos meus projetos audaciosos.

Minha Namorada:

Sandra, pelo carinho e paciência.

Meus Amigos:

Que felizmente são muitos e por mais que tentasse não conseguiria listá-los apenas nessa página.

AGRADECIMENTOS

Ao Professor Doutor Álvaro Geraldo Badan Palhares pela orientação durante o curso, onde transmitiu seus conhecimentos e sua experiência, pelo apoio durante as fases difíceis e pela amizade.

Aos grandes amigos e companheiros de laboratório Marconi Kolm Madrid e Marcus de Aguiar Dias, que me ajudaram do início a conclusão do curso.

Ao grande amigo Adson Ferreira da Rocha, pela inestimável amizade que me apoiou e ajudou durante o curso.

Um agradecimento especial ao amigo Eduardo Bonani, Técnico em Mecânica, que teve participação fundamental em meus trabalhos na parte mecânica.

Aos amigos Richard Gatinho Marques, Geovani Santo Prisinzano, Alcyr Silveira Franco e Marcelo Cortizo que participaram do projeto como alunos de iniciação científica.

Aos técnicos Yuri Evaristo Amorim e Marcelo Carrapato pelo apoio técnico.

A Técnica Regina do Laboratório de Circuito Impresso, pelo pronto atendimento e grande eficiência em seus serviços.

Ao CNPq pela concessão da bolsa de Mestrado.

A Faculdade de Engenharia Elétrica pelo financiamento do projeto.

A Divisão de Robótica do Centro Tecnológico para Informática pela transmissão de experiências em controle de robôs.

RESUMO

Neste trabalho é feito um estudo criterioso sobre os problemas relacionados com a operação de preensão mecânica de objetos e sua correlação com esta operação feita pela mão do homem, com o objetivo de projetar e construir garras mecânicas para utilização em manipuladores programáveis - Robôs.

Realiza-se uma análise cinemática e de forças baseado no sistema "screw" de coordenadas relacionando-se os movimentos e esforços envolvidos nos contatos entre o sistema prensor e o objeto a ser preso, bem como os tipos de contatos possíveis e necessários para uma preensão ideal.

Utilizando-se este tipo de ferramenta matemática inicia-se um estudo sobre as formas de controle de posição e força possíveis para se adotar numa garra de dois dedos, sendo um deles fixo, fazendo-se o projeto deste sistema de controle para a referida garra que foi construída no laboratório de Robótica do Departamento de Sistemas e Controle de Energia.

Os resultados obtidos para a análise de desempenho do controle de rigidez foram comparados com uma simulação do modelo matemático, que foi utilizado para permitir a escolha dos parâmetros de controle de acordo com uma especificação pré-definida.

A apresentação do trabalho é dividida em 5 capítulos cujos temas abordados são:

- Aspectos gerais sobre operação de preensão mecânica e a sua correlação com a feita pela mão humana - Capítulo 1.

- Análise cinemática e de forças em garras mecânicas visando a modelagem das mesmas. Este estudo baseou-se principalmente nos trabalhos de [Salisbury 1982-83] - Capítulo 2.

Estudo dos sistemas de posição e força em garra mecânicas existentes de força e rigidez, baseados nos trabalhos de [Salisbury 1982], [Hogan 1985], [Railbe r t & Craig 1981] e [Withney-1977].

Projeto e construção de uma garra mecânica de dois dedos paralelos sendo um fixo, acionada por um motor de corrente contínua e com realimentação de posição e força - Capítulo 4.

Resultados práticos da implementação descrita no capítulo 5, simulação do modelo dinâmico do sistema global e conclusões - Capítulo 5.

CAPÍTULO 1

1 ASPECTOS GERAIS SOBRE A OPERAÇÃO DE PREENSÃO

1.1	Introdução.....	01
1.2	Operação de Preensão por Mãos Humanas.....	03
1.2.1	Estrutura da mão humana.....	03
1.2.2	Função da mão.....	06
1.3	Caracterização dos Problemas Relacionados com Garras Mecânicas.....	12
1.4	Algumas Medidas de Qualidade em Garras Mecânicas.....	13
1.5	Escopo deste Trabalho.....	14

CAPÍTULO 2

2 ANÁLISE CINEMÁTICA E DE FORÇAS

2.1	Introdução.....	16
2.2	Sistema de Coordenadas "Screw" de Representação de Forças e Velocidades.....	17
2.3	Contatos.....	22
2.4	Efeitos de Contatos Múltiplos Entre Corpos.....	23
2.5	Transformação "Grip" (G).....	34
2.6	Sumário.....	36

CAPÍTULO 3

3 CONTROLE DE POSIÇÃO E FORÇA

3.1	Introdução.....	37
3.2	Estado da Arte em Controle de Posição e Força.....	38
3.3	Controle de Rigidez.....	46
3.4	Sumário.....	53

CAPÍTULO 4

4 PROJETO E IMPLEMENTAÇÃO DE UMA GARRA MECÂNICA DE DEDO FIXO

4.1	Introdução.....	54
4.2	Atuador do Tipo Motor cc.....	57
4.3	Sistema Mecânico.....	61
4.3.1	Apresentação Geral do Projeto.....	61
4.3.2	Análise Cinemática e de Forças.....	63
4.3.3	Descrição e Modelagem do Sistema Motor-Transmissão -Pinça Móvel.....	68
4.3.4	Descrição e Modelagem do Sistema Pinça Móvel-Objeto -Pinça Fixa.....	73
4.4	Acionamento: Recortador por MLP em Quatro Quadrantes.....	78
4.6	Controlador Digital.....	84
4.6.1	Descrição Geral do CPR1.....	84
4.6.2	Descrição da PCDD.....	85
4.6.2	Utilização do Sistema.....	89
4.7	Sensores.....	91
4.8	Sumário.....	91

CAPÍTULO 5

5 IMPLEMENTAÇÃO DO CONTROLE DE RIGIDEZ APLICADO NA GARRA DE DEDO FIXO CONSTRUÍDA

5.1	Introdução.....	93
5.2	Simulação do Modelo Dinâmico Da Garra com Controle de Rigidez.....	94
5.3	Implementação do Algoritmo de Controle de Rigidez.....	94
5.4	Resultados Experimentais.....	99
5.5	Análise dos Resultados Obtidos.....	107
5.6	Sumário.....	108

CAPÍTULO 6

6 CONCLUSÕES.....109

BIBLIOGRAFIA112

APÊNDICE A1 - A61

APÊNDICE B1 - B28

APÊNDICE C1 - C3

CAPÍTULO 1

ASPECTOS GERAIS SOBRE A OPERAÇÃO DE PREENSÃO

1.1 - Introdução

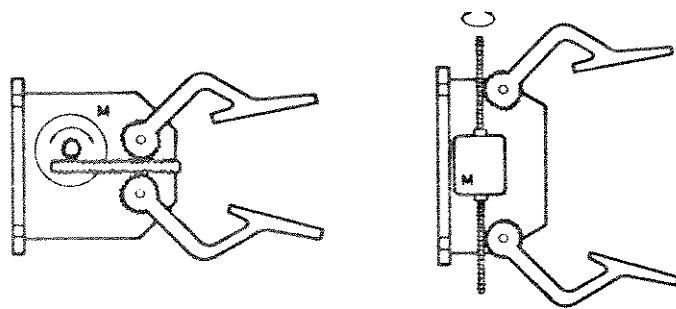
O projeto de garras mecânicas, de forma geral, tem como objetivo único o de desenvolver um mecanismo que realize um trabalho útil. Poder-se-ia relacionar o conceito de trabalho útil a uma tarefa específica, no entanto o que se busca com a utilização de garras mecânicas é uma maior flexibilidade na utilização de robôs com a mesma configuração em diversas tarefas.

Desta forma, uma garra mecânica pode ser considerada capaz de realizar um trabalho útil quando:

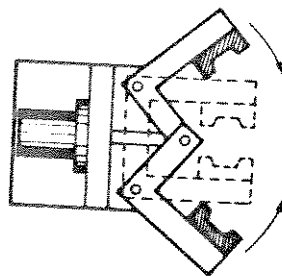
a) Imobilizar completamente um objeto preso pelo travamento de suas juntas e/ou;

b) Aplicar forças arbitrárias ou imprimir pequenos movimentos no objeto preso quando as juntas estão livres.

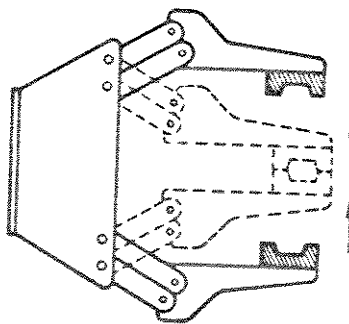
A grande maioria das garras mecânicas comerciais existentes são capazes de realizar a primeira classe de trabalho útil. Isso se deve ao fato de que esta é a primeira etapa para mudança de efetadores tipo ferramenta para tipo mão e também devido a classe de tarefas nas quais são utilizados robôs atualmente. A estrutura mecânica necessária para esta classe é mais simples como a garra de dedos pivotantes, dedos paralelos, etc. [Ferraté 1986]. Algumas dessas garras podem ser vistas na figura 1.1.



(a)



(b)



(c)

Figura 1.1 - Tipos de garras mecânicas.

- a) Pinças acionadas por motores
- b) Pinças de dedos pivotantes
- c) Pinças de dedos deslizantes

Para a realização da segunda classe de trabalho são utilizadas garras mecânicas articuladas ou dextras. Esse tipo de garra ainda está limitada aos laboratórios de pesquisa. Vários trabalhos são realizados nesta área como em [Hanafusa 1977], [Salisbury 1982] e [Cutkosky 1989].

Independentemente da classe para a qual a garra mecânica está sendo projetada, a pretensão é, na maioria das vezes, substituir a mão humana pela garra mecânica. No entanto, dada a complexidade estrutural e funcional das mãos humanas, essa tarefa é extremamente difícil.

Recentemente, os pesquisadores dessa sub-área da robótica sentiram a necessidade de conhecer a estrutura e função da mão humana com mais detalhes, para que desse estudo possa se obter conhecimentos mais aprofundados desse órgão tão complexo.

A próxima seção traz uma breve descrição desse órgão humano para que se possa sentir a complexidade de sua estrutura e funcionalidade que uma garra mecânica deve possuir.

1.2 - Operação de Prensão por Mãos Humanas

Este estudo é baseado nas pesquisas de J. Napier [Napier 1983], onde são feitos estudos da anatomia, função e evolução da mão do homem, e nos trabalhos, mais recentes, de M.R. Cutkosky [Cutkosky 1989], onde ele utiliza os conceitos fornecidos por Napier aplicados na escolha de preensões por garras mecânicas articuladas.

1.2.1 - Estrutura da mão humana

O esqueleto da mão humana é composto de 27 elementos (ossos). Todos se articulam com pelo menos outro elemento através de uma junta ou articulação sinovial. Os movimentos são controlados por músculos e ligamentos.

Das muitas espécies de juntas sinoviais, classificadas abaixo, de acordo com o tipo de movimento permitido, quase todas estão representadas na mão, cujo esqueleto e suas juntas sinoviais estão

mostrados na figura 1.2:

- Juntas Dobradiças: movimento em torno de um único eixo (as falanges articulando-se umas nas outras);
- Juntas Biaxiais: Tem dois eixos em ângulo reto de movimento (as falanges articulando-se com os metacarpos);
- Junta Selar: Tem as propriedades de uma junta biaxial e esférica (o metacarpo do polegar articulando-se com o trapézio, um dos ossos do carpo);
- Junta Plana: só permite o movimento de deslizamento entre duas superfícies (articulações entre os ossos adjacentes ao carpo).

Com essa estrutura complexa, a mão pode realizar uma série de movimentos:

- de flexão e extensão das falanges;
- lateral, pela junta, entre as falanges e os metacarpos;
- de afastamento de uma linha imaginária traçada com um meridiano que divida ao meio o dedo médio. Movimento este chamado de abdução, principal método de preensão de mamíferos não primatas;
- de aproximação dessa linha imaginária, chamado de adução.

Existe ainda na mão, um movimento muito importante no mecanismo de preensão chamado de movimento de oposição. Que é definido por *Napier* como:

"Oposição é o movimento pelo qual a superfície polpuda do polegar é colocada diretamente em contato com as almofadas terminais de um ou de todos os dedos restantes - ou diametralmente em oposição a estes."

Segundo *Napier*, sem o polegar a mão do homem recua sessenta milhões de anos em termos evolutivos, embora esse movimento não seja exclusivo dele, pois ocorre também nos macacos e grandes símios. No entanto, no homem esse movimento possibilita uma grande área de contato

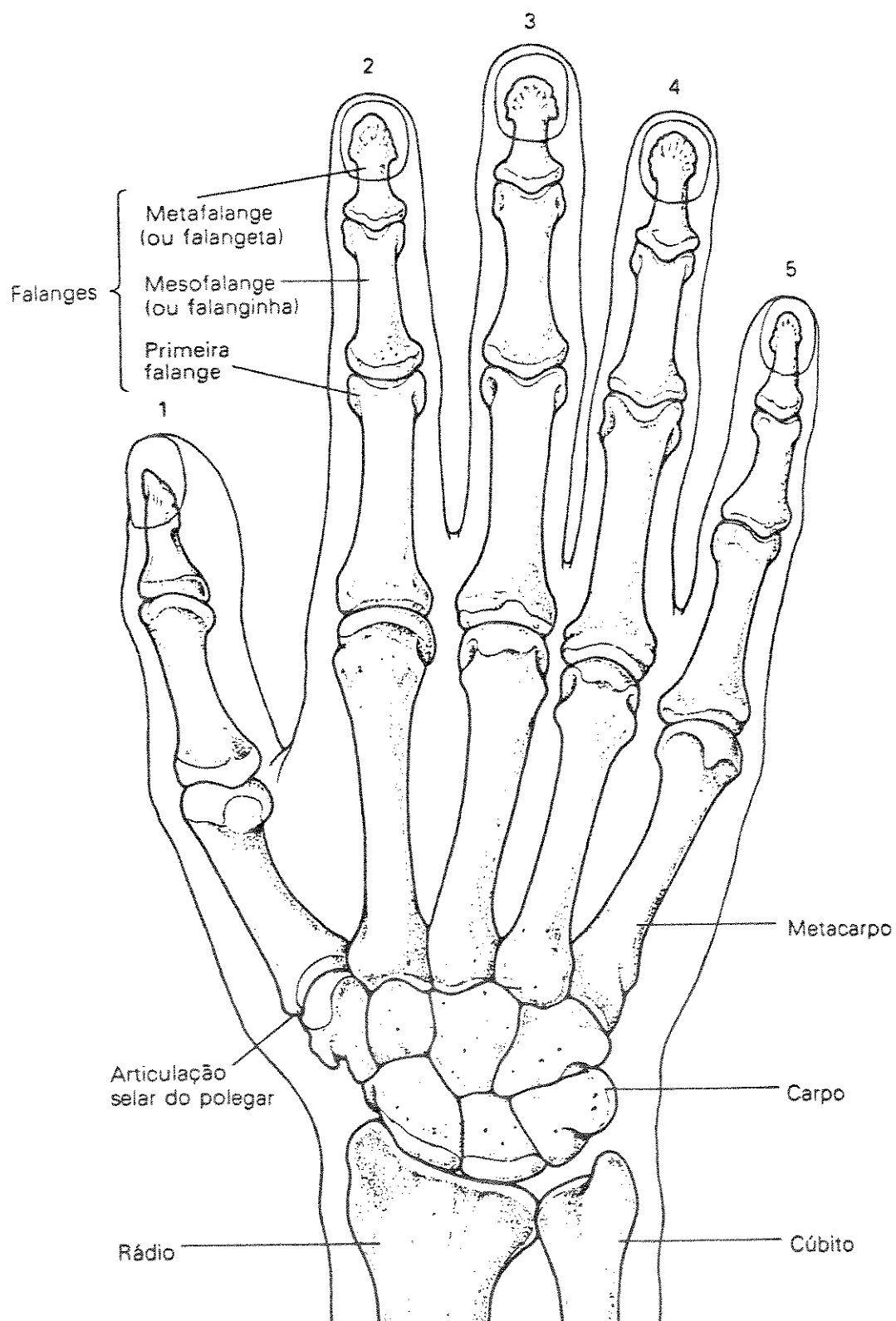


Figura 1.2 - Esqueleto da mão humana e suas juntas.

[Napier 1982]

entre as polpas dos dedos, o que leva a uma preensão mais eficaz e precisa de pequenos objetos.

Além da estrutura óssea e dos músculos e juntas responsáveis pelos movimentos, a mão possui pele, pelos, unhas e almofadas de gorduras.

A pele fornece uma proteção ao meio externo ao mesmo tempo em que faz o papel de sensor (tato), pois é nela que estão os terminais nervosos. É também na pele que estão as saliências papilares que além de ter um sistema de lubrificação, que mantém a pele úmida e adesiva, favorecendo a preensão, estimulam os terminais nervosos.

As unhas fornecem uma proteção para a ponta dos dedos, pois estes são de suma importância para manipulação de pequenos objetos e discriminação de textura.

As almofadas de gordura ou palmares, amortecem a pressão exercida pelos objetos sob preensão e defendem de lesões os tecidos mais internos.

Nas figura 1.3.a e 1.3.b pode ser observado a grande complexidade da mão humana. Descrita por Napier como:

"Retirada a luva de pele da mão e revelada sua estrutura interna começa-se a perceber o que significa "multum in parvo". Num espaço incrivelmente pequeno, tendões, músculos, nervos, artérias e veias estão empilhados como sardinhas em lata..."

1.2.2 - Função da mão

A mão é capaz de realizar dois tipos de operações:

- Preênsil: onde um objeto é agarrado por uma ação de apertar ou pinçar entre os dedos e/ou palma;
- Não-preênsil: empurrar, levantar, bater e apontar com os dedos, girar, etc.

Dentro de movimentos preênséis existem um número enorme que a mão pode executar. No entanto, segundo Napier existem apenas dois

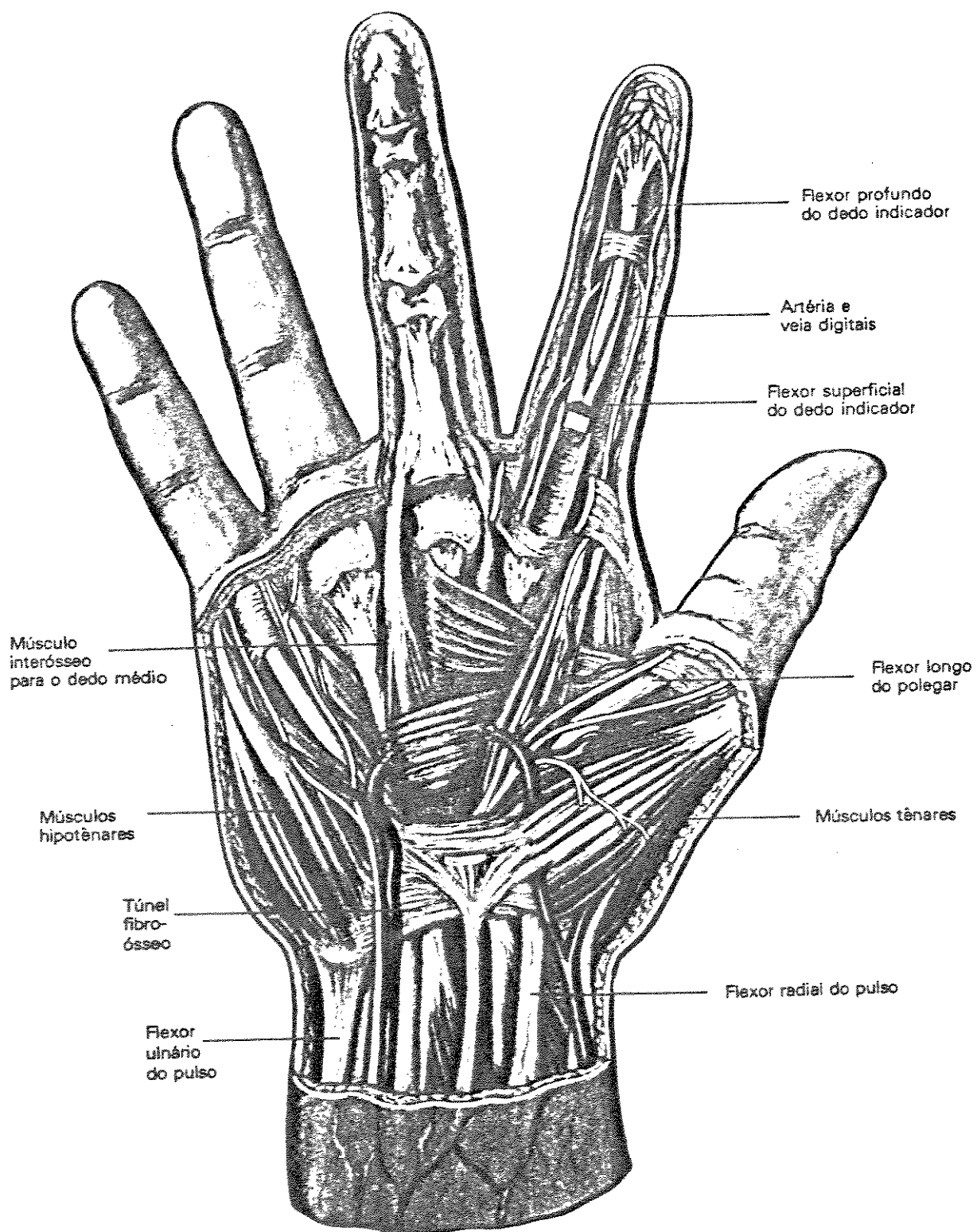


Figura 1.3.a - Estrutura da mão humana. [Napier 1982]
Vista palmar dissecada.

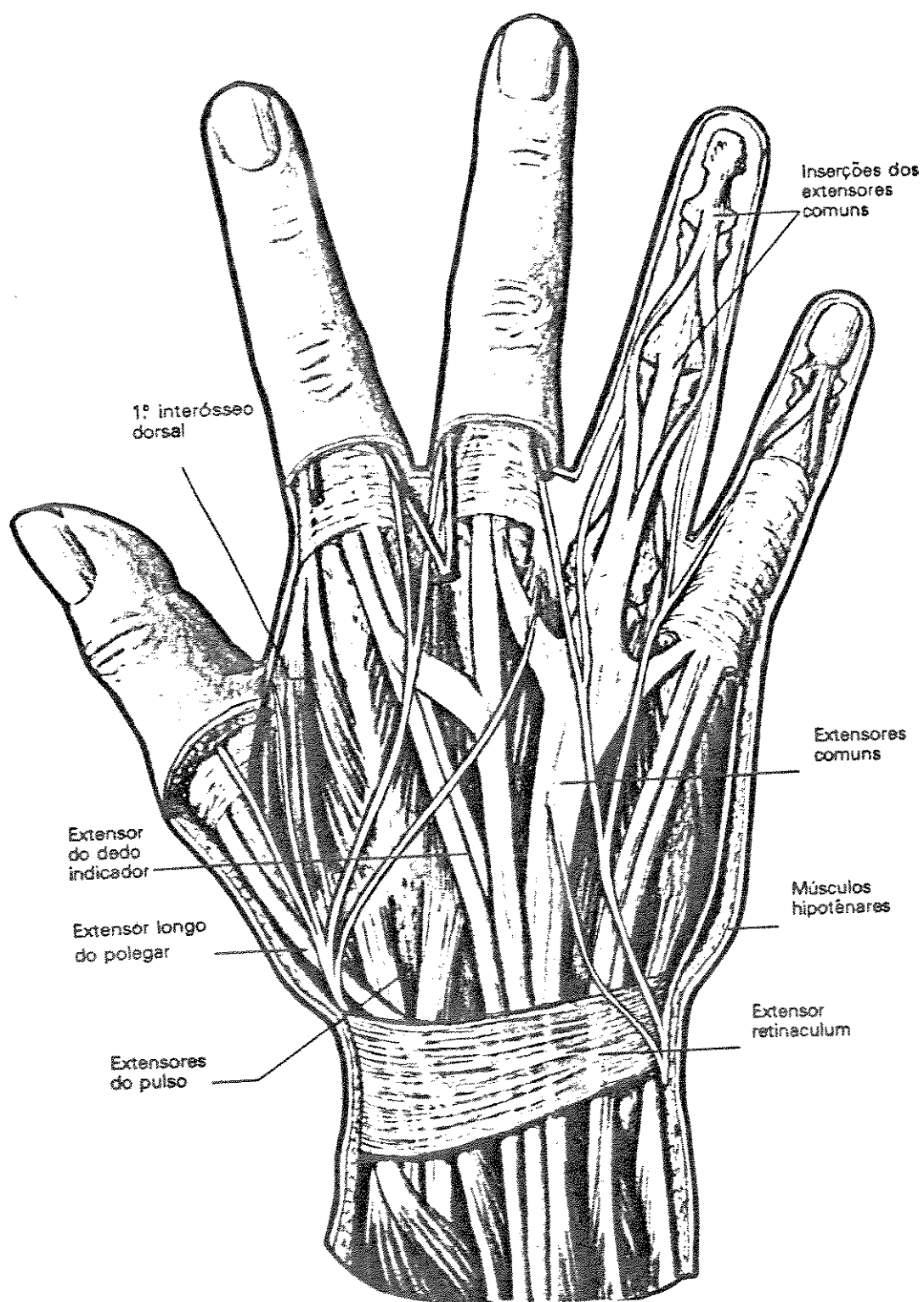


Figura 1.3.b - Estrutura da mão humana [Napier 1982]
Vista dorsal dissecada.

padrões principais e dois padrões subsidiários de preensão:

- Padrões principais:
 - Preensão de precisão;
 - Preensão de força.
- Padrões subsidiários:
 - Preensão gancho;
 - Preensão tesoura.

A preensão de precisão é executada entre as pontas dos dedos, como a preensão que é realizada no momento em que estamos ajustando uma lâmpada no bocal, figura 1.4.a.. Este tipo de preensão é empregado quando uma tarefa exige delicadeza e precisão de manipulação.

A preensão de força é executada entre a superfície dos dedos e a palma da mão, como a preensão realizada no momento em que queremos atarrachar a lâmpada no bocal com firmeza, figura 1.4.b.. Este tipo de preensão é empregado quando uma tarefa exige força para sua execução.

É bom salientar que geralmente as operações de preensão utilizam os dois tipos apresentados ao mesmo tempo, só que um deles é de consideração secundária.

A preensão gancho é um tipo de preensão que é função dos flexores dos dedos, onde os nós das juntas estão dobradas e travadas. É a preensão usada quando segura-se uma mala pesada.

A preensão tesoura é o tipo de preensão em que o objeto preendido é agarrado entre os lados dos dedos. É o tipo de preensão usado para segurar um cigarro entre os dedos.

Poder-se-ia considerar que o tipo de preensão usado é função apenas da atividade a ser exercida na tarefa e não depende do formato ou tamanho do objeto a ser manuseado, no entanto, essa generalização passa a não ser válida no caso dos objetos serem muito grandes ou muito pequenos, ou quando o formato do objeto for feito sob encomenda.

Cutkosky [Cutkosky 1989], realizando estudos de preensão em tarefas de pequenos grupos de usinagem e montagem por homens, obteve uma taxionomia parcial, apresentada na figura 1.5. Essa taxionomia

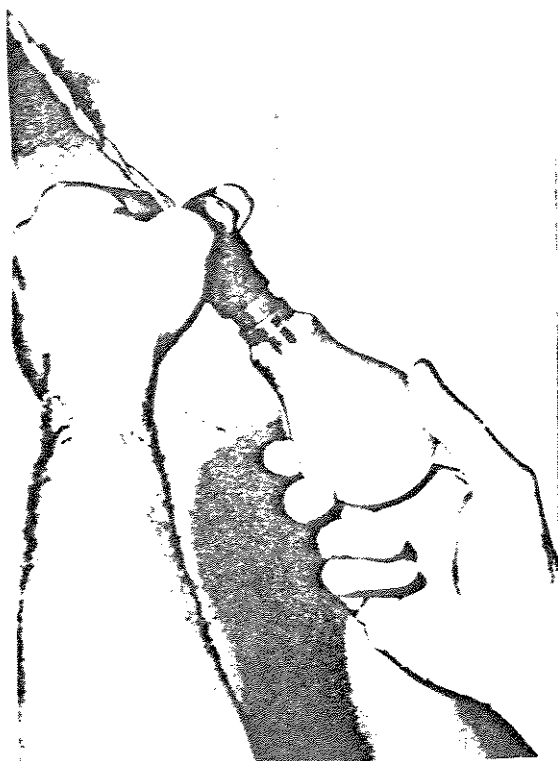


Figura 1.4.a - Exemplo de Preensão de Precisão

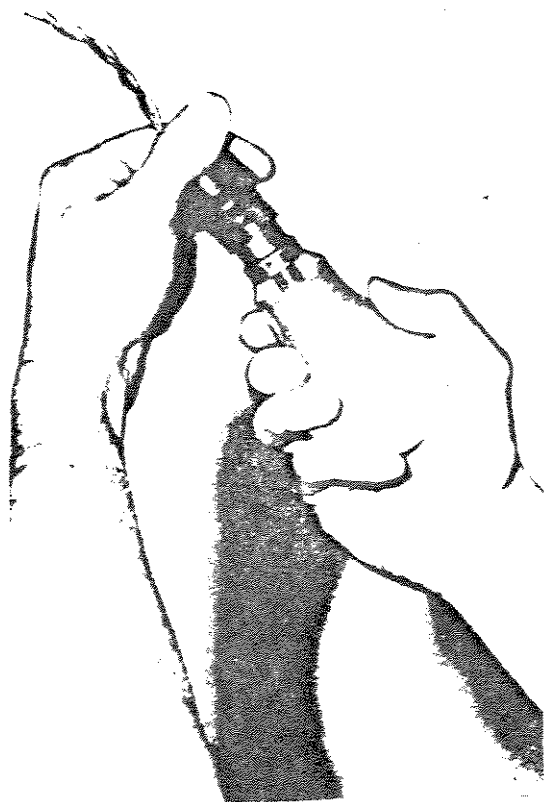


Figura 1.4.b - Exemplo de Preensão de Força

inicia com os dois padrões principais sugeridos por Napier e depois, forma-se uma árvore hierárquica de preensões. Quando move-se para baixo na árvore, detalhes da tarefa e geometria do objeto tornam-se igualmente importantes para que na análise final, ambas requisições da tarefa e forma do objeto formem uma região central na determinação da preensão.

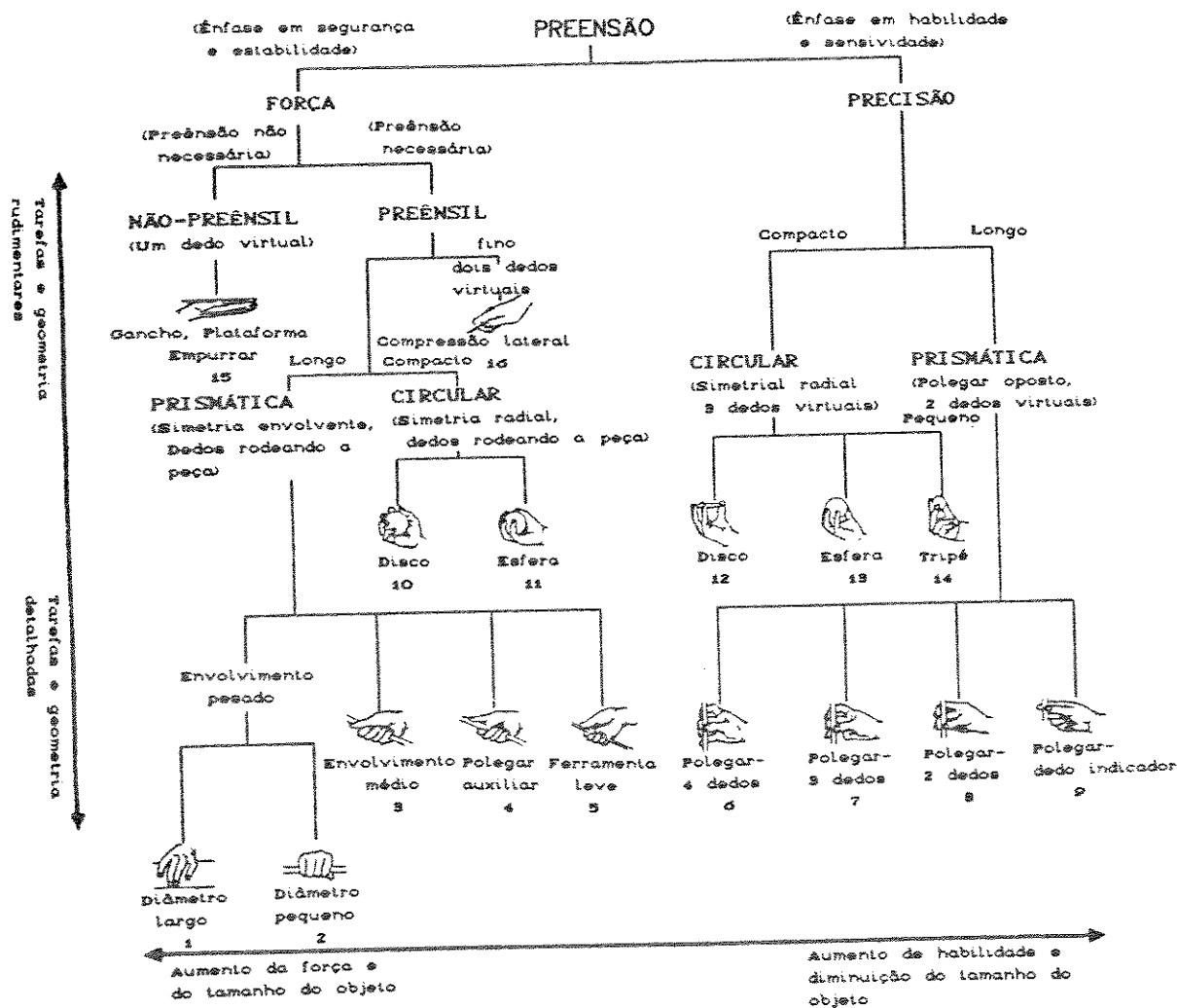


Figura 1.5 - Taxionomia Parcial de Preensões Humanas
[Cutkosky 1989]

1.3 - Caracterização dos Problemas Relacionados com Garras Mecânicas

Como pode ser visto através do estudo da mão humana, o projeto e análise de garras mecânicas exigirá um estudo em diversas áreas da mecânica e eletrônica para a geração dos comandos dos movimentos e sensoreamento.

De um modo geral, o estudo de garras mecânicas articuladas pode ser resumido em três grandes áreas, sem perda de precisão:

a)Análise Cinemática e de Forças: Estuda as estruturas mecânicas e em quais condições estas satisfazem requisitos necessários para execução de alguma tarefa útil. Neste estudo, além das ferramentas matemáticas utilizadas no estudo de manipuladores mecânicos de cadeia aberta (Jacobiano, modelo cinemático direto e inverso, etc.), são utilizadas outras ferramentas matemáticas para análise de mecanismos de cadeia fechada;

b)Controle de Posição e Força: Desde que já se possua uma estrutura mecânica que seja capaz de realizar algum trabalho útil, é necessário que exista um sistema de controle para comandá-la. Esta fase engloba diversas áreas de estudo, tais como: análise do modelo dinâmico, controle de servomecanismos, atuadores, sistemas de acionamento, sistemas digitais e dos algoritmos de controle;

c)Escolha Automática de Preensões: Pode-se dizer que é o problema de mais alto nível hierárquico, pois só depois de dominada as áreas anteriores é que será possível criar um sistema para "escolher" qual o tipo de preensão que a garra irá utilizar para prender um objeto, ou mesmo manipulá-lo. É o módulo que dá inteligência à garra mecânica, geralmente está ligada a estudos de inteligência artificial.

Para tornar o problema de garras mecânicas mensurável existem diversas medidas de qualidade. Como a nomenclatura desta área ainda encontra-se em fase de definição, um resumo das medidas mais comumente utilizadas faz-se necessário. Isso é feito na seção

seguinte.

1.4 - Algumas Medidas de Qualidade em Garras Mecânicas

Baseado em vários modelos analíticos de preensão e manipulação de objetos por garras mecânicas, Cutkosky [Cutkosky 1989] resumiu um número de medidas de qualidade para essas operações, que de forma simples podem ser enunciadas da seguinte maneira:

- **Flexibilidade:** Medida que determina qual deslocamento (deformação) ocorrerá no objeto preso quando um conjunto de forças é imposto pela garra. Seu inverso é denominado de **Rigidez**. É função da configuração da preensão, do servomecanismo da junta e das flexibilidades da estrutura.

- **Conectividade:** É o número de parâmetros independentes que são necessários para especificar completamente a posição e orientação do objeto preendido em relação a "palma" da garra.

- **Força de Encerramento:** Uma preensão satisfaz uma força de encerramento se, quando as juntas dos dedos estão travadas e existem forças externas que mantêm o contato entre a garra e o objeto, este não pode se mover sem deslizamento.

- **Forma de Encerramento:** Existe uma forma de encerramento ou uma restrição cinemática completa quando podem existir forças e momentos externos aplicados em qualquer direção sobre o objeto preso e este não se mover, sendo que as juntas dos dedos da garra estejam travadas.

- **Preensão Isotrópica:** É uma medida de qualidade que determina se uma dada configuração de preensão permite que as juntas dos dedos da garra apliquem precisamente forças e momentos sobre o objeto preso.

- **Força Internas:** São forças que existem internamente ao objeto preso devido as interações entre suas partículas. Suas componentes não produzem qualquer variação na sua quantidade de movimento [Alonso & Finn 1978]. Formalmente, são as soluções homogêneas da equação de equilíbrio do objeto preso.

- **Manipulabilidade:** É uma medida que determina se os dedos podem imprimir movimentos arbitrários sobre o objeto.

- **Resistência de deslizamento:** Determina as forças e momentos aplicados sobre o objeto preendido antes que este comece a deslizar.

- **Estabilidade:** Uma preensão é estável se ela retorna para sua condição inicial após uma perturbação por forças e momentos externos.

1.5 - Escopo deste Trabalho

Como pode ser visto, o estudo de garras mecânicas envolve uma vasta área de conhecimento, inviabilizando assim um esgotamento deste assunto em um único trabalho. Assim, procurou-se limitar este trabalho aos seguintes objetivos principais:

- Estudar o estado da arte em modelagem e controle de garras mecânicas articuladas;

- Aplicar os conceitos estudados para projetar e implementar um controlador para uma garra mecânica viável de ser construída nas condições existentes do laboratório;

Para que estes objetivos fossem alcançados, o trabalho foi organizado nas seguintes etapas:

a) Estudo da análise cinemática e de forças de garras mecânicas visando a modelagem das mesmas. Este estudo foi baseado nos

trabalhos de Salisbury [Salisbury 1982 e 1983] e está disposto no capítulo 2;

b) Estudo de controle de posição e força existentes. São abordados o controle de posição, de força, e de rigidez. Este estudo está baseado nos trabalhos de [Withney 1977], [Salisbury 1982], [Hogan 1985], e [Vassoughi 1987]. Esta parte está descrita no capítulo 3;

c) Projeto e construção de uma garra de dedo fixo acionada por motor de corrente contínua, com realimentação de posição e força. Capítulo 4;

d) Simulação e implementação do controle de rigidez. A simulação e o algoritmo de controle são apresentados no capítulo 5;

e) Resultados experimentais, discussões e conclusões sobre o trabalho. Capítulo 5.

CAPÍTULO 2

ANÁLISE CINEMÁTICA E DE FORÇAS

2.1 - Introdução

Este capítulo tem como objetivo introduzir conceitos relacionados a análise cinemática e de forças utilizados no estudo de garras articuladas.

Na análise cinemática e de forças são dados conceitos relacionados com contatos e seus efeitos na restrição de movimentos e é feita com base no sistema de coordenadas "screw", cuja teoria está resumidamente descrita na seção 2.2 e foi baseado nos trabalhos de [Ohwovoriole & Roth 1981], [Lipkin & Duffy 1982] e [Salisbury 1982].

Através dessa análise, dos conceitos de mobilidade e conectividade e fazendo algumas considerações sobre a restrição de movimentos, são listados os mecanismos que satisfazem uma condição necessária para restrição completa de objetos presos por garras articuladas. Essa análise é feita com base nos estudos de Salisbury [Salisbury 1982] e [Salisbury 1983]. Depois são feitas considerações de ordem prática que levam a uma análise através de uma álgebra aproximada para verificação de restrição completa de objetos.

Durante a análise cinemática e de forças é realizado um estudo em uma transformação linear denominada de "Grip" (G), que relaciona forças e velocidades resultantes no corpo preso com forças e velocidades nos contatos dos dedos. É uma transformação similar ao Jacobiano em mecanismos de cadeia aberta.

É importante salientar que a análise das restrições impostas por contatos entre a garra e o meio é estritamente válida somente para

o instante em consideração, mas como o conjunto de restrições não mudam drasticamente, esta análise pode ser estendida para pequenos movimentos.

2.2 - Sistema de Coordenadas "Screw" de Representação de Forças e Velocidades

Uma forma eficiente e objetiva para descrever o movimento espacial de um corpo rígido e das forças que atuam sobre este devido a restrições complexas é utilizando o sistema "screw" de representações de forças e velocidades, desenvolvidos por R.S. Ball em seu trabalho "A treatise on the Theory of Screws", em 1900. Este sistema enfatiza a dependência fundamental de sua representação sobre linhas geométricas.

A seguir é feito um resumo das definições envolvidas no conceito de "screw".

Definição 2.1 : "Screw" (contorção)

Um "screw" é definido por uma linha reta no espaço com um grau associado, o "pitch" (p). É representado por um vetor sextuplo $\underline{s} = [S_1 \ S_2 \ S_3 \ S_4 \ S_5 \ S_6]^T$, que também pode ser representado através de seus componentes da seguinte forma, que será utilizada neste trabalho por ser a mais comum: $\underline{s} = (S_1, S_2, S_3, S_4, S_5, S_6)$. Um "screw" com seu eixo definido pelo vetor coseno diretor $\underline{l}$, de componentes (L_1, L_2, L_3) , passando pelo ponto $\underline{r}$, de componentes (R_1, R_2, R_3) , e possuindo grau p , tem as seguintes componentes :

$$S_1 = L_1;$$

$$S_2 = L_2;$$

$$S_3 = L_3;$$

$$S_4 = K_1 - p \cdot L_1;$$

$$S_5 = K_2 - p \cdot L_2;$$

$$S_6 = K_3 - p \cdot L_3.$$

onde:

$\underline{k} = [\underline{r} \times \underline{l}] = [K_1 \ K_2 \ K_3]^T$, momento de linha em torno da origem do sistema de referência.

então:

$$\underline{s} = \begin{bmatrix} 1 \\ \vdots \\ \underline{r} \times \underline{l} - p \cdot \underline{l} \end{bmatrix}$$

O grau do "screw" é definido por:

$$p = \frac{S_1 \cdot S_4 + S_2 \cdot S_5 + S_3 \cdot S_6}{S_1^2 + S_2^2 + S_3^2}$$

A magnitude do "screw" é dado por:

$$\begin{aligned} (S_1^2 + S_2^2 + S_3^2)^{1/2} & , \text{ se } p \neq \infty \text{ e} \\ (S_4^2 + S_5^2 + S_6^2)^{1/2} & , \text{ se } p = \infty \end{aligned}$$

As operações de multiplicação escalar e adição vetorial são válidas para "screws" infinitesimais, assim dois "screws" são considerados linearmente independentes se for possível achar escalares a_1 e a_2 diferentes de zero, tais que:

$$a_1 \cdot \underline{s}_1 + a_2 \cdot \underline{s}_2 = 0$$

Definição 2.2: "Twist" (eixo de giro)

Em um movimento infinitesimal de um corpo no espaço existe associado uma única linha, o eixo "twist" (movimento helicoidal, em torno do qual ele rotaciona e ao longo do qual ele translada. Este movimento é chamado de "twist" e é identificado por um vetor sextuplo, representado através de suas componentes sob forma:

$$\underline{t} = (T_1, T_2, T_3, T_4, T_5, T_6), \text{ onde:}$$

T_1, T_2 e T_3 são componentes da velocidade angular $\underline{w}$ e
 T_4, T_5 e T_6 são componentes da velocidade linear $\underline{v}$.

com $\underline{w}$ em torno do vetor $\underline{l}$ e $\underline{v}$ velocidade de um ponto fixado no corpo e referenciado na origem do sistema pelo vetor $\underline{r}$.

Então:

$$\underline{t} = \begin{bmatrix} \underline{w} \\ \dots \\ \underline{v} \end{bmatrix} = w \cdot \begin{bmatrix} \underline{l} \\ \dots \\ \underline{r} \times \underline{l} - p \cdot \underline{l} \end{bmatrix}$$

O grau do "twist" é dado por:

$$p = \frac{\underline{w} \cdot \underline{v}}{\underline{w} \cdot \underline{w}}$$

desta forma, se:

$p = 0 \Rightarrow$ movimento de rotação pura e
 $p = \infty \Rightarrow$ movimento de translação pura.

A magnitude de um "twist" é dada por:

$$\|\underline{w}\|, \text{ se } p \neq \infty \text{ e}$$

$$\|\underline{v}\|, \text{ se } p = \infty.$$

Definição 2.3: "Wrench" (eixo de arranco)

Um conjunto de forças e momentos atuantes em um corpo é equivalente a uma única força atuando ao longo de um eixo, denominado eixo "wrench" (arranco), e um momento exercido em torno desse eixo. Essa combinação da força e momento é chamada de "wrench" e pode ser identificada por um vetor sextuplo $\underline{w} = (W_1, W_2, W_3, W_4, W_5, W_6)$, onde:

W_1, W_2 e W_3 são componentes da força resultante $\underline{f}$ na direção $\underline{l}$ e

W_4, W_5 e W_6 são componentes do momento resultante $\underline{m}$.

então:

$$\underline{w} = \begin{bmatrix} \underline{f} \\ \underline{m} \end{bmatrix} = f \cdot \begin{bmatrix} \underline{l} \\ \underline{r} \times \underline{l} - p \cdot \underline{l} \end{bmatrix}$$

O grau do "wrench" é definido por:

$$p = \frac{\underline{f} \cdot \underline{m}}{\underline{f} \cdot \underline{f}}$$

se

$p = 0 \Rightarrow$ força pura

$p = \infty \Rightarrow$ momento puro

A magnitude de um "wrench" é dada por:

$\|\underline{f}\|$, se $p \neq \infty$

$\|\underline{m}\|$, se $p = \infty$.

Quando a magnitude de um "screw", "twist" ou "wrench" é igual a um, eles são conhecidos como "screw", "twist" ou "wrench" unitários. Como "twists" e "wrenches" geralmente são associados a uma magnitude diferente de um, pode-se relacionar um "twist" ou um "wrench" como uma magnitude atuando ao longo de um "screw", com as unidades convenientes.

Os sistemas "twists" e "wrenches" pertencem a um espaço vetorial sextuplo, podendo-se, portanto, aplicar qualquer conceito relacionado a espaços vetoriais.

Definição 2.4: "Screws" Recíprocos

Um outro conceito utilizado neste sistema de representação é o de "screws" recíprocos. Dois "screws", $\underline{s}_1 = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6)$ e

$\underline{s}_2 = (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6)$, são recíprocos se o coeficiente virtual, dado por:

$$c.v. = \alpha_1. \beta_4 + \alpha_2. \beta_5 + \alpha_3. \beta_6 + \alpha_4. \beta_1 + \alpha_5. \beta_2 + \alpha_6. \beta_3$$

for igual a zero. Este coeficiente virtual pode ser interpretado fisicamente como trabalho realizado por um "wrench"

$$\underline{w} = \begin{bmatrix} \underline{f} \\ \underline{m} \end{bmatrix}$$

aplicado a um corpo em movimento com um "twist"

$$\underline{t} = \begin{bmatrix} \underline{w} \\ \underline{v} \end{bmatrix}$$

então se não existe trabalho realizado o "twist" e o "wrench" são ditos recíprocos.

É importante ressaltar que para representar todos os movimentos possíveis ou todos esforços a que um corpo está sujeito, sempre existe um número mínimo de "screws" necessários. Assim, é interessante definir o conceito de ordem de um sistema "screw".

Definição 2.5: Ordem de um sistema "screw"

A ordem de um sistema "screw", n , é igual ao número de bases requeridas para defini-lo. E a ordem do sistema "screw" recíproco é igual a $6-n$.

A teoria "screw" é bastante extensa e foram mencionadas somente algumas definições que dizem respeito diretamente a este trabalho, sendo que para um aprofundamento maior podem ser consultados as referências citadas no início desta seção.

2.2 - Contatos

Outro conceito importante para a realização do estudo de preensão de objetos é o conceito de contatos, pois a preensão depende fundamentalmente do tipo e quantidade de contatos existentes entre o objeto a ser preso e o instrumento preensor.

Definição 2.6: Contato

Dado dois corpos se tocando, A e B, figura 2.1, Salisbury [Salisbury 1982] define contato como sendo um conjunto de pontos adjacentes onde os toques ocorrem sobre uma área contínua. Duas áreas disjuntas de contato são consideradas dois contatos distintos.

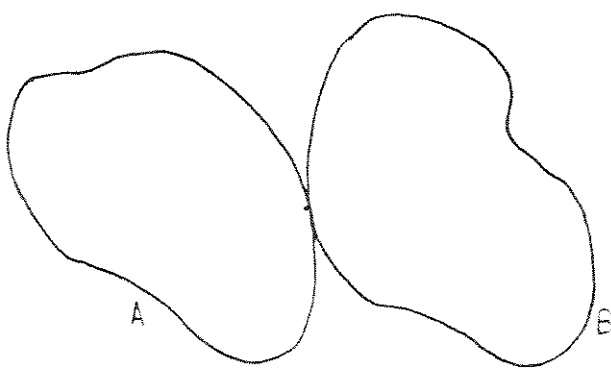


Figura 2.1 - Dois corpos em contato.

A limitação do movimento relativo entre os corpos depende de quatro fatores relacionados com as áreas de contato: Forma, localização, orientação e atrito.

Com relação ao atrito entre superfícies, alguns trabalhos não o consideram, confiando apenas na restrição estrutural do mecanismo, como em [Ohwovoriole & Roth 1981]. Outros o consideram como ativo ou não ativo e levam isso em consideração na restrição dos graus de liberdade de movimento, isso é feito em [Salisbury 1982]. E uma terceira abordagem é a de considerar o atrito, incluindo uma análise

das condições sob as quais ele é ativo, isso é feito em [Ohwovoriote 1987]. Neste trabalho será abordado o segundo caso.

Por questões de modelamento, os contatos são classificados em quatro tipos distintos, de acordo com a superfície de toque: Pontual, linear, planar e "toque suave". Na tabela 2.1 pode ser verificado os efeitos desses contatos em relação aos graus de liberdades. Verifica-se que o efeito depende da consideração do atrito ser ou não ativo.

O contato tipo toque suave ("soft touch"), comporta-se como um contato pontual com atrito, exceto que a área de contato é grande o bastante para resistir a momentos em torno da normal ao contato. Desta forma, é um contato com dois graus de liberdade.

Tabela 2.1 - Efeitos de Contatos na Restrição de Movimentos. [Salisbury 1982]

Tipos de contato	G. D. L	
	Sem atrito	Com atrito
Contato Pontual	5	3
Contato Linear	4	1
Contato Planar	3	0
Toque Suave	-	2

Na figura 2.2 são listados os conjuntos de bases unitárias "twists" e "wrenches" para os quatro tipos de contatos. As liberdades permitidas para cada contato é igual a ordem do sistema "twist" associado.

2.3 - Efeitos de Contatos Múltiplos Entre Corpos

Até o momento falou-se apenas nos efeitos de um só contato sobre um corpo. Na análise de vários contatos atuando em um corpo usa-se uma versão da fórmula de Grübler apresentada por Salisbury [Salisbury 1982], para determinar mobilidades e conectividades do

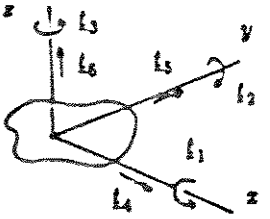
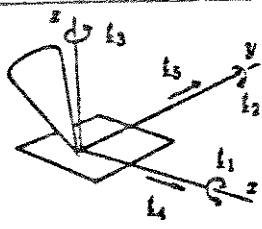
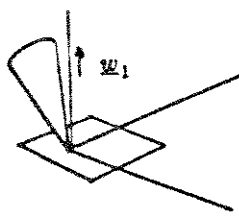
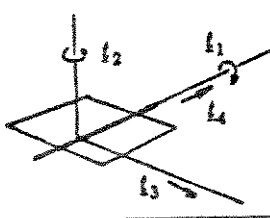
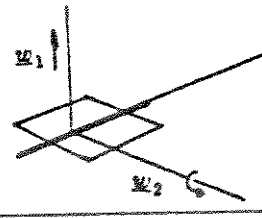
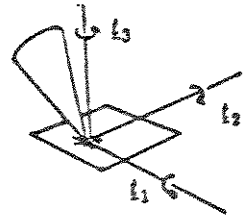
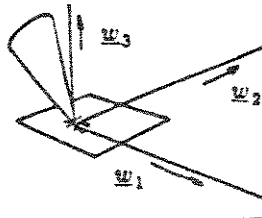
Tipos de contato	Sistema Twist	Sistema Wrench
Sem contato 6		
	$t_1 = (1, 0, 0, 0, 0, 0)$ $t_2 = (0, 1, 0, 0, 0, 0)$ $t_3 = (0, 0, 1, 0, 0, 0)$ $t_4 = (0, 0, 0, 1, 0, 0)$ $t_5 = (0, 0, 0, 0, 1, 0)$ $t_6 = (0, 0, 0, 0, 0, 1)$	Sistema wrench nulo
Contato pontual 5 sem atrito		
	$t_1 = (1, 0, 0, 0, 0, 0)$ $t_2 = (0, 1, 0, 0, 0, 0)$ $t_3 = (0, 0, 1, 0, 0, 0)$ $t_4 = (0, 0, 0, 1, 0, 0)$ $t_5 = (0, 0, 0, 0, 1, 0)$	$w_1 = (0, 0, 1, 0, 0, 0)$
Contato linear 4 sem atrito		
	$t_1 = (0, 1, 0, 0, 0, 0)$ $t_2 = (0, 0, 1, 0, 0, 0)$ $t_3 = (0, 0, 0, 1, 0, 0)$ $t_4 = (0, 0, 0, 0, 1, 0)$	$w_1 = (0, 0, 1, 0, 0, 0)$ $w_2 = (0, 0, 0, 1, 0, 0)$
Contato pontual 3 com atrito		
	$t_1 = (1, 0, 0, 0, 0, 0)$ $t_2 = (0, 1, 0, 0, 0, 0)$ $t_3 = (0, 0, 1, 0, 0, 0)$	$w_1 = (1, 0, 0, 0, 0, 0)$ $w_2 = (0, 1, 0, 0, 0, 0)$ $w_3 = (0, 0, 1, 0, 0, 0)$

Figura 2.2 - Representação dos contatos através do sistema "screw". [Salisbury 1982]

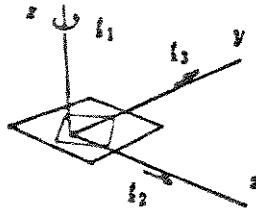
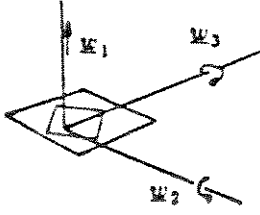
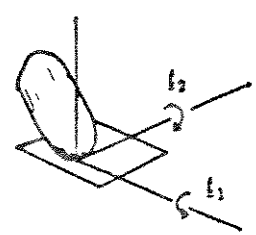
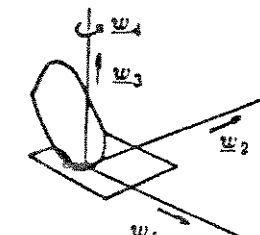
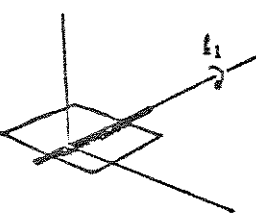
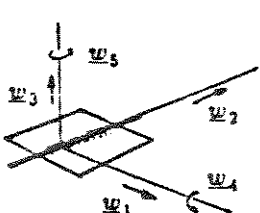
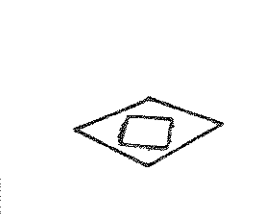
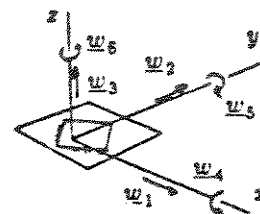
Contato planar 3 sem atrito	 $\begin{aligned} t_1 &= (0, 0, 1, 0, 0, 0) \\ t_2 &= (0, 0, 0, 1, 0, 0) \\ t_3 &= (0, 0, 0, 0, 1, 0) \end{aligned}$	 $\begin{aligned} w_1 &= (0, 0, 1, 0, 0, 0) \\ w_2 &= (0, 0, 0, 1, 0, 0) \\ w_3 &= (0, 0, 0, 0, 1, 0) \end{aligned}$
Contato toque suave	 $\begin{aligned} t_1 &= (1, 0, 0, 0, 0, 0) \\ t_2 &= (0, 1, 0, 0, 0, 0) \end{aligned}$	 $\begin{aligned} w_1 &= (1, 0, 0, 0, 0, 0) \\ w_2 &= (0, 1, 0, 0, 0, 0) \\ w_3 &= (0, 0, 1, 0, 0, 0) \\ w_4 &= (0, 0, 0, 0, 0, 1) \end{aligned}$
Contato linear 1 com atrito	 $t_1 = (0, 1, 0, 0, 0, 0)$	 $\begin{aligned} w_1 &= (1, 0, 0, 0, 0, 0) \\ w_2 &= (0, 1, 0, 0, 0, 0) \\ w_3 &= (0, 0, 1, 0, 0, 0) \\ w_4 &= (0, 0, 0, 1, 0, 0) \\ w_5 &= (0, 0, 0, 0, 0, 1) \end{aligned}$
Contato planar 0 com atrito	 Sistema twist nulo	 $\begin{aligned} w_1 &= (1, 0, 0, 0, 0, 0) \\ w_2 &= (0, 1, 0, 0, 0, 0) \\ w_3 &= (0, 0, 1, 0, 0, 0) \\ w_4 &= (0, 0, 0, 1, 0, 0) \\ w_5 &= (0, 0, 0, 0, 1, 0) \\ w_6 &= (0, 0, 0, 0, 0, 1) \end{aligned}$

Figura 2.2 - Continuação.

sistema garra-objeto.

Nesta análise observa-se apenas o número de graus de liberdade das juntas do mecanismo e dos contatos deste com o meio.

Definição 2.7: Mobilidade

A mobilidade M de um sistema cinemático é definido como o número de parâmetros necessários para especificar completamente a posição de todos os corpos em um sistema no instante de observação.

A versão da fórmula de Grübler para cálculo da mobilidade é:

$$M \geq \sum f_i + \sum g_j - 6.L \quad (2.1)$$

ou

$$M' \geq \sum g_j - 6.L$$

onde:

M = mobilidade do sistema;

M' = mobilidade do sistema com juntas travadas;

f_i = graus de liberdade da i -ésima junta;

g_j = graus de liberdade do j -ésimo contato;

L = número de laços independentes do sistema.

Definição 2.8: Conectividade

Do conceito de mobilidade pode-se derivar o conceito de conectividade, C , que é o número de parâmetros necessários para especificar completamente a posição relativa de dois corpos no instante de observação. Este conceito é muito importante, pois, na manipulação de objetos a preocupação é com o movimento relativo entre a garra e o objeto preendido. C' representa a conectividade do sistema com juntas travadas.

Com esses conceitos pode-se determinar as estruturas que realizam trabalho útil, ou seja, as estruturas que são capazes de :

- Aplicar forças arbitrárias ou imprimir pequenos movimentos no objeto preendido quando as juntas estão livres;
- Imobilizar completamente um objeto preso pelo travamento de suas juntas.

A primeira condição implica que a conectividade do sistema garra-objeto deve ser igual a seis ($C = 6$). E a segunda condição implica em conectividade com juntas travadas igual ou menor que zero ($C' \leq 0$).

Assim, analisando a conectividade pode-se enumerar os mecanismos que satisfazem as condições acima impostas. No entanto, é necessário utilizar o critério de que um dedo com três juntas pode tocar um objeto de oito maneiras diferentes. A figura 2.3 ilustra essas maneiras e um número para identificá-las.

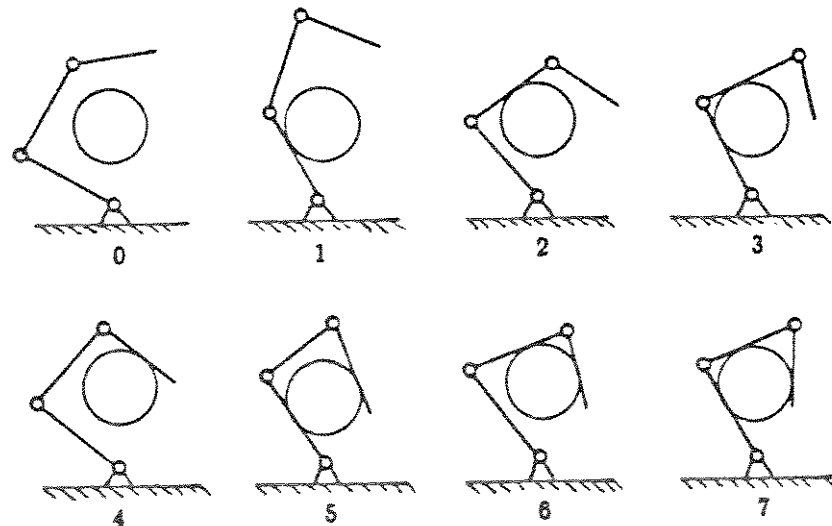


Figura 2.3 - Identificação das maneiras que um dedo com três juntas pode tocar um objeto.

Assim, se um mecanismo com três dedos e três juntas por dedo puder tocar um objeto, seu código será dado pelos números correspondentes de cada tipo de contato de cada dedo. Um exemplo pode ser visto na figura 2.4.

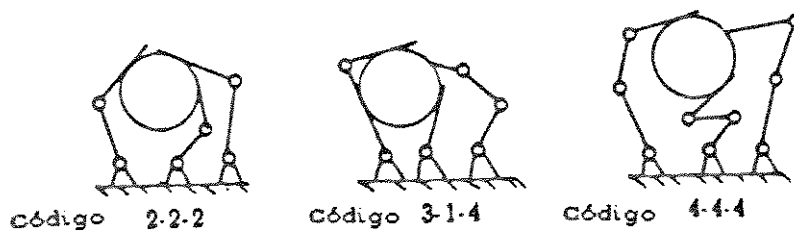


Figura 2.4 - Exemplo de enumeração de mecanismos.

Supondo que todos contatos entre garra e objeto apresentam o mesmo número de graus de liberdade, então o número de estruturas a serem consideradas são um total de seiscentas. Destes apenas trinta e nove satisfazem as condições de conectividade impostas. Esses mecanismos são listados na tabela 2.2.

Para determinar se uma preensão particular em um corpo impõe restrições suficientes para imobilizá-lo completamente deve ser feito uma análise nos movimentos permitidos ou nas forças que podem ser aplicadas.

Do ponto de vista de movimento, quando um corpo está sob a ação de vários contatos o movimento resultante ou o sistema "twist" disponível para o corpo será a interseção dos sistemas "twists" de cada contato.

Do ponto de vista de forças, quando vários contatos atuam sobre o corpo o sistema "wrench" resultante que pode ser aplicado será a união de todos sistemas "wrenches" dos contatos. Algebricamente o "wrench" resultante, $\underline{w}$, é dado por:

$$\underline{w} = [w_1 \ w_2 \ \dots \ w_n] \quad \underline{c} = \underline{W} \cdot \underline{c} \quad (2.2)$$

onde:

w_i : são as coordenadas dos "wrenches" principais de cada sistema "wrench", são vetores colunas de dimensão seis;
 $\underline{c}$: é um vetor n-dimensional arbitrário de coeficientes correspondentes as intensidades dos "wrenches";
 $\underline{W}$: é uma matriz $6 \times n$ formada pelos vetores coluna, cujos elementos são as coordenadas dos "wrenches" principais.

Se os sistemas "twists" são denominados de S_i e os sistemas "wrenches" de S_i' (sistema recíproco), para imobilizar completamente um corpo é necessário e suficiente que:

$$S_1 \cap S_2 \cap \dots \cap S_n = \emptyset \quad (2.3)$$

ou

$$S_1' \cup S_2' \cup \dots \cup S_n' = \mathbb{R}^6 \quad (2.4)$$

Sendo assim, quando se deseja determinar se um objeto está completamente imobilizado por um conjunto de contatos, deve-se utilizar as equações (2.3) ou (2.4), a que melhor se adequar. Para situações mais complexas a equação (2.2) deve ser utilizada.

Aplicando esta condição sobre os mecanismos listados na tabela 2.2, estes deverão observar as seguintes regras geométricas:

a) Para contatos com 3 G.D.L. (pontuais com atrito e planares sem atrito), devem existir:

- Três contatos pontuais com atrito não colineares ou;
- Dois contatos pontuais e um contato planar sem atrito, desde que a linha que liga os dois pontos não seja normal ao plano do contato planar ou;
- Dois contatos planares sem atrito não paralelos e um contato pontual com atrito ou;
- Três contatos planares sem atrito, dois dos quais não paralelos e se duas das três linhas de interseção dos planos não tiverem a mesma direção.

b) Para contatos com 4 G.D.L. (Contatos lineares sem atrito), são necessários pelo menos três contatos lineares sem atrito que satisfaçam, dois a dois, as seguintes condições:

- Os contatos lineares não devem ser paralelos e;
- Os planos que contém cada contato linear não devem ser paralelos e;

Tabela 2.2 - Mecanismos que satisfazem a condição de conectividade ($C=6$ e $C' \leq 0$). [Salisbury 1982]

Tipos de Mecanismo	Total de Juntas	M,M'	C,C'
Contatos de 3 G.D.L.			
4-4-0	6	6,0	6,0
4-4-4	9	6,-3	6,-3
Contatos de 4 G.D.L.			
2-2-2	6	6,0	6,0
4-2-2	7	7,0	6,0
4-4-2	8	8,0	6,0
4-4-4	9	9,0	6,0
Contatos de 5 G.D.L.			
7-7-0	6	6,0	6,0
7-3-1	6	6,0	6,0
7-5-1	7	7,0	6,0
7-6-1	7	7,0	6,0
7-7-1	7	6,-1	6,-1
7-3-2	7	7,0	6,0
7-5-2	8	8,0	6,0
7-6-2	8	8,0	6,0
7-7-2	8	7,-1	6,-1
3-3-3	6	6,0	6,0
5-3-3	7	7,0	6,0
6-3-3	7	7,0	6,0
7-3-3	7	6,-1	6,-1
7-4-3	8	8,0	6,0
5-5-3	8	8,0	6,0
6-5-3	8	8,0	6,0
7-5-3	8	7,-1	6,-1
6-6-3	8	8,0	6,0
7-6-3	8	7,-1	6,-1
7-7-3	8	6,-2	6,-2
7-5-4	9	9,0	6,0
7-6-4	9	9,0	6,0
7-7-4	9	8,-1	6,-1
5-5-5	9	9,0	6,0
6-5-5	9	9,0	6,0
7-5-5	9	8,-1	6,-1
6-6-5	9	9,0	6,0
7-6-5	9	8,-1	6,-1
6-6-5	9	9,0	6,0
7-6-5	9	8,-1	6,-1
7-7-5	9	7,-2	6,-2
6-6-6	9	9,0	6,0
7-6-6	9	8,-1	6,-1
7-7-6	9	7,-2	6,-2
7-7-7	9	6,-3	6,-3

- A normal comum aos contatos linear não deve ser paralela a linha de interseção dos dois planos.

c) Para contatos com 5 G.D.L. (contatos pontuais sem atrito), são necessários pelo menos seis contatos para imobilizar completamente um objeto. No entanto, achar regras geométricas torna-se uma tarefa muito complexa. Desta forma, torna-se mais indicado verificar se pelo menos seis dos "wrenches" impostos pela garra sobre o objeto são linearmente independentes, o que equivale verificar se o posto da matriz de coordenadas dos "wrenches" principais, W (equação (2.2)), é igual a seis. Ainda assim, existem certas formas que não podem ser imobilizadas com apenas seis "wrenches" independentes, como uma esfera.

Através da análise de todas essas considerações pode-se concluir que o mecanismo que melhor se enquadra é o tipo 4-4-4 com 3 G.D.L. por contato, pois, neste projeto as juntas redundantes contribuem para uma preensão mais segura. Isso pode ser observado na conectividade com as juntas travadas $C' = -3$, onde o valor negativo implica em restrições redundantes que permitem controlar forças internas para manter as restrições impostas pelo atrito. [Salisbury 1983]

Ainda resta fazer uma consideração a mais para a restrição completa de um objeto, que é a de considerar que alguns "wrenches" atuam apenas em um sentido. Desta forma, mesmo que a matriz das coordenadas dos "wrenches" principais da preensão possua posto igual a seis, pode ocorrer que para apenas um subconjunto de perturbações o corpo permaneça completamente restringido. Então uma análise algébrica aproximada pode permitir saber se uma dada preensão pode ou não restringir um objeto.

Para um "wrench" arbitrário w ser aplicado no objeto sem que exista perturbação do objeto preso, deve existir um vetor c que satisfaça a equação (2.2), onde os p primeiros vetores coluna da matriz W representam os "wrenches" que possuem um único sentido,

$$W = [w_1 \ w_2 \ \dots \ w_p \ w_{p+1} \ \dots \ w_n]$$

Se W possui posto igual a seis:

$$\underline{c} = W^{-1} \cdot \underline{w} \quad (2.5)$$

Assim, se $n = 6$, da equação (2.5), os p primeiros elementos de $\underline{c}$ devem ser positivos, pois eles serão os coeficientes dos "wrenches" que possuem um único sentido. Se isso não acontecer ocorrerá quebra do contato ou um deslize do objeto preso, não sendo válida mais a equação (2.5).

No caso de existirem um número de "wrenches" maior que seis deve-se considerar os seguintes casos:

a) $n = 7$:

As soluções da equação (2.4) terão a forma:

$$\underline{c} = \underline{c}_p + \lambda \cdot \underline{c}_h \quad (2.6)$$

onde:

$\underline{c}_p$ = solução particular;

$\underline{c}_h$ = solução homogênea;

λ = variável arbitrária que determina a magnitude da força interna.

Se todos p primeiros elementos de $\lambda \cdot \underline{c}_h$ forem positivos, para qualquer valor de $\underline{c}_p$, um λ pode ser achado que fará todos os p primeiros elementos de $\underline{c}$ positivos e que satisfazem a equação (2.2).

b) $n > 7$:

Isso implica que se o número de "wrenches" atuando no objeto for maior do que sete, então a solução homogênea terá $(n-6)$ variáveis livres que podem ser escolhidas para fazerem os p primeiros elementos de $\underline{c}$ positivos. Assim, existirá mais que um conjunto de forças

internas independentes.

Para exemplificar, considere um corpo conectado por dois contatos do tipo toque suave com seus "wrenches" correspondentes indicados na figura 2.5.

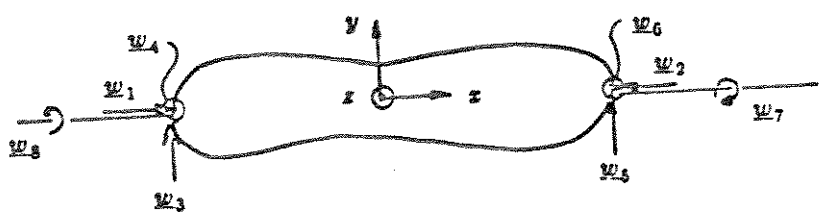


Figura 2.5 - Exemplo de um corpo conectado por dois contatos tipo toque suave.

Pode-se observar que w_1 e w_2 são forças de contato as quais devem ser mantidas positivas para assegurar o contato e permitir que os "wrenches" w_3 , w_4 , w_5 , w_6 e w_7 , que dependem do atrito permaneçam ativos. Se $p=2$ então:

$$\underline{g}h = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \cdot \lambda_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \cdot \lambda_2$$

Então, os p primeiros elementos de $\underline{g}h$ podem ser feitos arbitrariamente positivos e a preensão mostrada na figura 2.6 pode restringir completamente o objeto. A preensão possui duas forças

internas independentes.

2.4 - Transformação "Grip" (G)

As relações entre força e velocidades em um sistema garra-objeto podem ser reunidas na transformação linear "grip", G. Onde, G, é uma matriz quadrada de dimensão igual ao número de "wrenches" de contato dos dedos com o objeto. Esta matriz de transformação necessita ser recalculada somente quando a preensão muda significativamente.

A matriz $(G^{-1})^T$ pode ser obtida da seguinte forma:

$$(G^{-1})^T = \begin{bmatrix} -\frac{W}{c_{1,h}^T} - \\ \vdots \\ \frac{c_{n-6,h}^T}{c_{n-6,h}^T} \end{bmatrix} \tag{2.7}$$

onde:

W = matriz de coordenadas dos "wrenches" principais da preensão
 $c_{i,h}$ = vetor base da solução homogênea da equação (2.2)

Se os $(n-6)$ $c_{i,h}$'s são linearmente independentes e o espaço nulo de W tiver dimensão $(n-6)$, $(G^{-1})^T$ será inversível.

Definido:

$$\mathcal{F} = [f_x \ f_y \ f_z \ m_x \ m_y \ m_z \ \lambda_1 \ \dots \ \lambda_{n-6}]^T$$

$$\underline{E} = [C_1 \ C_2 \ \dots \ C_n]^T$$

onde:

$\mathcal{F}$ = força resultante sobre o objeto;
 $\underline{E}$ = vetor intensidade dos contatos "wrenches";
 f_x, f_y e f_z = são as componentes da força do "wrench" resultante sobre

o objeto;

m_x, m_y e m_z = são as componentes do momento do "wrench" resultante sobre o objeto;

λ_i = magnitude das forças internas;

C_i = intensidade dos "wrenches".

Pode ser derivada a seguinte relação entre forças nos contatos e forças no objeto:

$$\mathcal{F} = (G^{-1})^T \cdot E \quad (2.)$$

Por esta relação pode-se determinar que forças externas estão atuando no objeto preso e que forças internas são suportadas pelos dedos através da medição de forças e momentos atuando nos dedos, ou para determinar as intensidades dos "wrenches" para que se pressione o objeto com uma determinada força resultante.

Pode-se também derivar uma relação de velocidades definindo-se:

$$\begin{aligned} \gamma &= [v_x \ v_y \ v_z \ w_x \ w_y \ w_z \ \gamma_1 \ \gamma_2 \ \dots \ \gamma_{n-6}]^T \\ \underline{V} &= [D_1 \ D_2 \ \dots \ D_n]^T \end{aligned}$$

onde:

γ = velocidades resultantes do objeto;

$\underline{V}$ = vetor intensidade dos "twists" de cada contato;

v_x, v_y e v_z = componentes da velocidade linear resultante de um ponto no objeto;

w_x, w_y e w_z = componentes da velocidade angular resultante do objeto;

γ_i = velocidades virtuais resultantes da deformação;

D_i = intensidade dos "twists" existentes.

e a relação,

$$\gamma = G \cdot \underline{V} \quad (2.9)$$

permite obter a velocidade resultante no objeto. Dado que se tem as magnitudes de cada "twist", ou dado que se queira uma velocidade resultante pode-se determinar qual deve ser a velocidade nos contatos dos dedos.

2.5 - Sumário

Foi visto neste capítulo uma série de conceitos relacionados a análise cinemática e de forças. Através dessa análise pode-se observar as diversas considerações para que um objeto seja completamente restringido por uma garra. Deve ser enfatizado que a abordagem adotada não é única, no entanto fornece um meio relativamente simples de análise.

Com esses conceitos, aliados ao controle de posição e esforços que serão vistos no próximo capítulo, pode-se fazer um estudo mais rigoroso e detalhado no projeto de mecanismos para garras articuladas.

CAPÍTULO 3

CONTROLE DE POSIÇÃO E FORÇA

3.1 - Introdução

Como foi visto no capítulo anterior é possível construir mecanismos capazes de prender completamente um objeto ou até mesmo que sejam capazes de manipular o objeto preso. No entanto, um mecanismo é uma estrutura passiva que necessita de um sistema de controle e um sistema de acionamento para poder comandá-lo.

O sistema da garra mecânica a ser controlado é complexo, pois, sempre existirá uma interação física de contato entre o mecanismo e o objeto, formando assim um mecanismo de cadeia fechada.

Em tarefas onde existe um contato físico entre o manipulador, o objeto e o meio, ocorre uma mudança brusca no modelo dinâmico do sistema, pois, no espaço livre as forças que predominam são as forças inerciais, gravitacionais, centrípeta e de coriolis. Já após o contato, a força que predomina é a força de contato, que depende da rigidez do meio e da estrutura do mecanismo. O grande problema está, portanto, na transição de um movimento no espaço livre para um espaço restrito, ou seja, de um meio onde a rigidez era zero para um onde a rigidez é da ordem de 10^4 a 10^8 N/m, dependendo dos materiais envolvidos.

Muitos trabalhos são feitos no controle de tarefas que requerem contato entre o manipulador e o meio, no entanto, apenas alguns são extensivos para o uso em garras articuladas. A seguir será feito uma breve revisão bibliográfica dos trabalhos envolvidos com

esta problemática.

3.2 - Estado da Arte em Controle de Posição e Força

Hanafusa e Asada [Hanafusa & Asada 1977] introduziram o conceito de estabilidade de uma preensão. Por este conceito uma preensão é estável se quando a posição relativa entre a mão e o objeto desvia de uma certa situação, uma força de restituição é gerada pelos dedos para que a posição relativa retorne a situação original. De acordo com este princípio básico foram criados critérios para a condição de estabilidade para uma determinada preensão. Com a evolução deste estudo eles propuseram um esquema de controle para uma mão robótica com dedos elásticos [Hanafusa & Asada 1978]. Nesse esquema a elasticidade dos dedos eram dados pela inserção de molas entre as pontas dos dedos e os motores. O diagrama esquemático da mão utilizada é apresentada na figura 3.1.

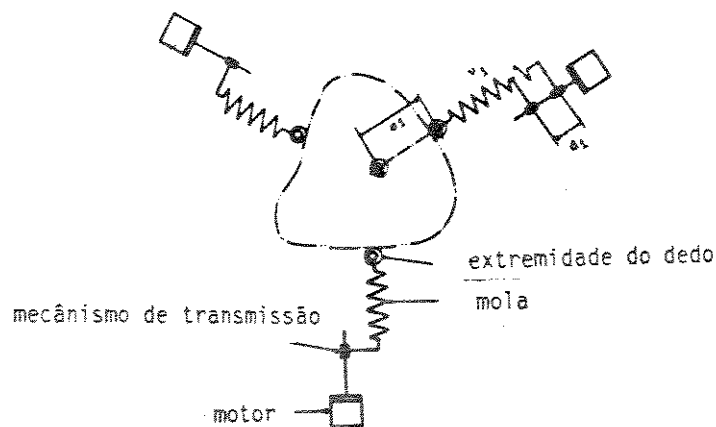


Figura 3.1 - Diagrama esquemático da mão robótica proposta em [Hanafusa & Asada 1978]

onde:

- α_i : deslocamento do i -ésimo dedo;
- d_i : deslocamento do i -ésimo motor;
- v_i : deformação da i '-ésima mola.

O princípio fundamental do esquema de controle proposto para controlar ativamente o sistema de dedos elásticos é dado pela equação (3.1):

$$f_i = K_i \cdot (\sum b_{ij} \cdot \alpha_j + c_i), \quad i=1,2 \text{ e } 3 \quad (3.1)$$

onde:

f_i : força do i -ésimo dedo;
 K_i : constante da i -ésima mola;
 α_i : deslocamento do i -ésimo dedo;
 b_{ij} e c_i : constantes arbitrárias que são ajustadas para obter desejadas características na preensão.

De acordo com esse princípio foi feito uma análise do sistema de preensão e projeto do sistema de controle voltados para aplicações em processos de montagem.

Whitney [Whitney 1977] estudou o problema do controle de força de um braço usando sensores de força em um servossistema para melhorar seu desempenho. Foi usado uma matriz de admitância para especificar velocidades do braço em resposta a forças medidas. É basicamente um controle de velocidade que possui um efeito resultante de controle das forças de contato da tarefa. A matriz de admitância pode ser construída para expressar estratégias de montagem para cada tarefa particular. Este tipo de controlador é conhecido por controle de amortecimento.

O diagrama de blocos da figura 3.2 apresenta um servossistema linear com realimentação de força. Este controlador utiliza a velocidade como variável de entrada, e a partir da medida de força aplicada esta referência é modificada para se obter um comando de velocidade. A realimentação de posição, devidamente dimensionada, pode melhorar a resposta dinâmica do sistema.

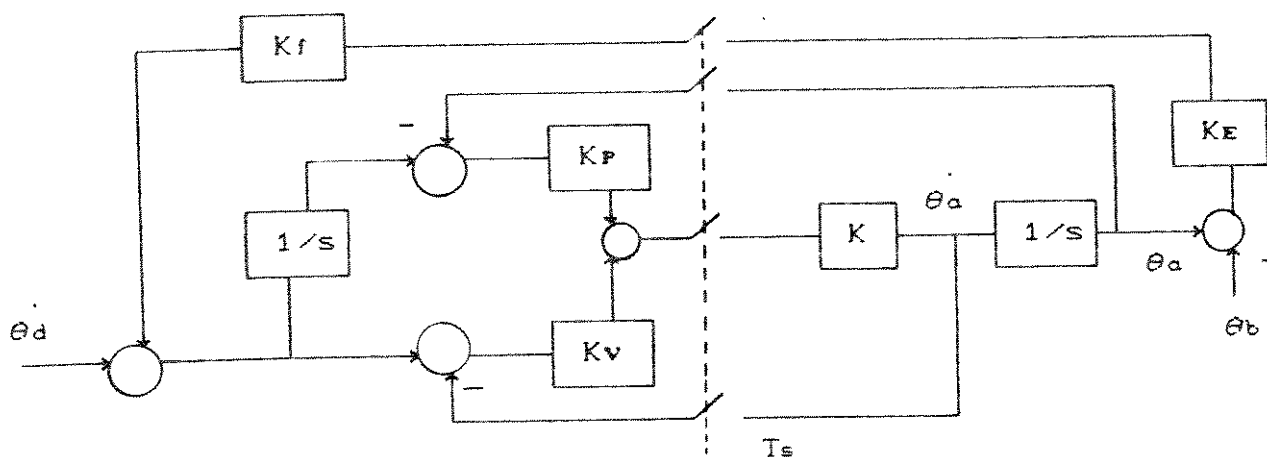


Figura 3.2- Diagrama de blocos de um servossistema linear com realimentação de força. [Whitney 1977]

onde:

K_f, K_v, K_P : ganhos das realimentações de força, velocidade e posição.
 K_E : Coeficiente de rigidez do sensor e do meio;
 $\dot{\theta}_d$: Velocidade comandada;
 $\dot{\theta}_a, \theta_a$: Velocidade e posição Medida;
 θ_b : Posição do meio envolvido.

Salisbury [Salisbury 1980] descreve um método para controle ativo de rigidez de um manipulador em coordenadas cartesianas. Por este método o programador pode especificar a rigidez dos três movimentos translacionais e dos três movimentos rotacionais no sistema de coordenadas cartesiano localizado na garra. Este procedimento foi extendido para controle de garras mecânicas articuladas em [Salisbury & Craig 1981] e [Salisbury 1982]. Este tipo de controle será visto com detalhes posteriormente, pois este será adotado para controle da garra projetada do capítulo 4.

Raibert e Craig [Raibert & Craig 1981] elaboraram uma estratégia de controle denominada controle híbrido de posição e força. Esta estratégia permite ao programador selecionar certos eixos para

serem controlados em posição e os restantes controlados em força. A seleção dos eixos de controle de forças ou posição no efetuador depende das restrições impostas pela tarefa. Uma análise feita por Mason [Mason 1981] fornece uma base teórica para essa seleção em tarefas simples. Na figura 3.3 é apresentado a organização conceitual do controlador híbrido.

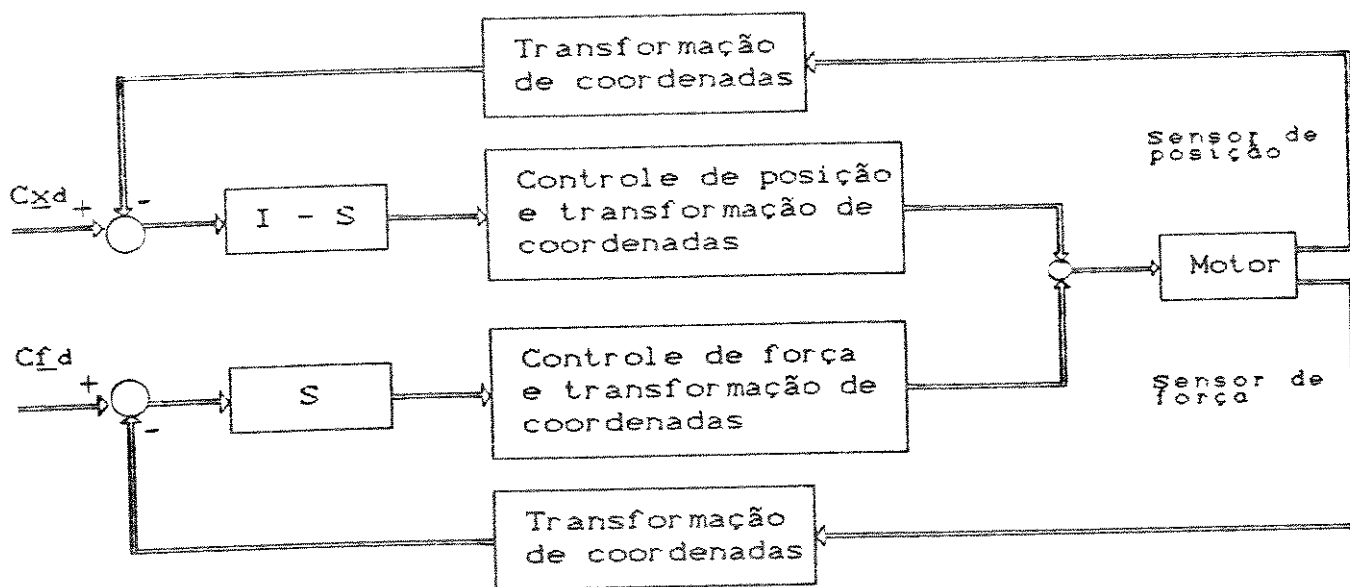


Figura 3.3 - Organização conceitual do controlador híbrido
[Raibert & Craig 1981]

onde:

c_{xd} : Trajetória da posição desejada;

c_{fd} : Trajetória da força desejada;

S : Matriz selecionadora do controle de posição ou força.

As transformações de coordenadas transformarão posições e forças no espaço das juntas para o espaço cartesiano ou vice-versa. Se for utilizado um sensor de força de pulso a transformação de coordenadas para forças do espaço de juntas para o espaço cartesiano não existirá, eliminando assim uma operação matricial. Essa simplificação é muito importante, visto que uma grande desvantagem no

uso deste tipo de controle é a grande quantidade de operações matriciais que deve ser efetuadas.

A matriz S é diagonal e só pode ter elementos iguais a 1, para selecionar o controle de força, ou 0, para selecionar controle de posição. Sua dimensão é $n \times n$, onde n é o número de juntas do manipulador.

Hogan [Hogan 1985] fez considerações de que somente o controle de posição ou força isoladamente são inadequados ao controle de tarefas que possuem restrições físicas, devendo portanto, considerar o comportamento dinâmico da interação entre o manipulador e o meio. Este trabalho considera que o manipulador deve ter um comportamento de impedância. É uma generalização dos controles de rigidez e de amortecimento, pois estes são parâmetros da impedância. Em [Cabrera 1990] é analisado esse tipo de controle para robôs com juntas elásticas. Estudou-se o desempenho de uma junta com controle de impedância quando a flexibilidade da mesma não é desprezível. Essa flexibilidade da junta utilizada baseia-se no fenômeno da torção na transmissão do torque entre o atuador e a carga.

A figura 3.4 apresenta o controlador de impedância no espaço cartesiano para um robô com várias juntas.

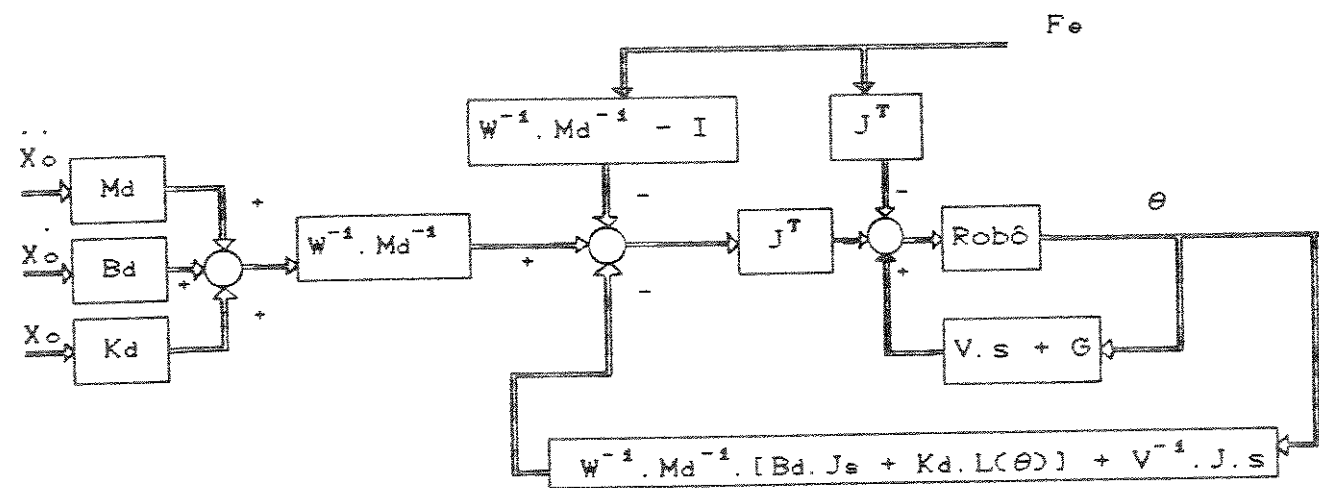


Figura 3.4 - Diagrama de blocos do controlador de impedância para uma robô com várias juntas. [Cabrera 1991]

onde:

$\ddot{X}_0, \dot{X}_0, X_0$: Referência de posição, velocidade e aceleração;
 F_e : Vetor de forças medidas no espaço cartesiano;
 θ : Vetor de posições das juntas;
 M_d : Matriz de inércia desejada;
 B_d : Matriz de amortecimento desejada;
 K_d : Matriz de rigidez desejada;
 M : Matriz de inércia;
 V : Vetor de termos de coriolis e centrífuga;
 G : Vetor de termos de gravidade;
 $W = J.M^{-1}.J^T$.

Parker e Paul [Parker & Paul 1987] fazem uma análise quantitativa sobre a força de impacto do dedo de uma garra mecânica com o objeto, baseado em valores obtidos experimentalmente, e descreve um método de controle da velocidade de impacto para reduzir esta força. Este método utiliza um controle ótimo/dual - Tempo mínimo/proporcional de um dedo com velocidade final igual a zero. O controlador utilizado em uma garra com acionamento pneumático é apresentado na figura 3.5.

A servoválvula pneumática (controle proporcional) segura o dedo na máxima posição aberta antes da mão ser comandada para prender um objeto. Quando este comando é enviado do controlador, o solenóide (controle bang-bang) é ativado no lugar da servo-válvula. Quando o dedo aproxima-se do objeto e o sinal de proximidade é recebido, o solenóide é chaveado para o estado de normalmente fechada e a servoválvula é novamente ativado. Um controle de força é usado para levar o dedo até entrar em contato com o objeto. Como a velocidade é baixa, existirão baixas forças de contato. Uma vez estando o dedo em contato com o objeto, ele pode ser controlado tanto por força como por posição.

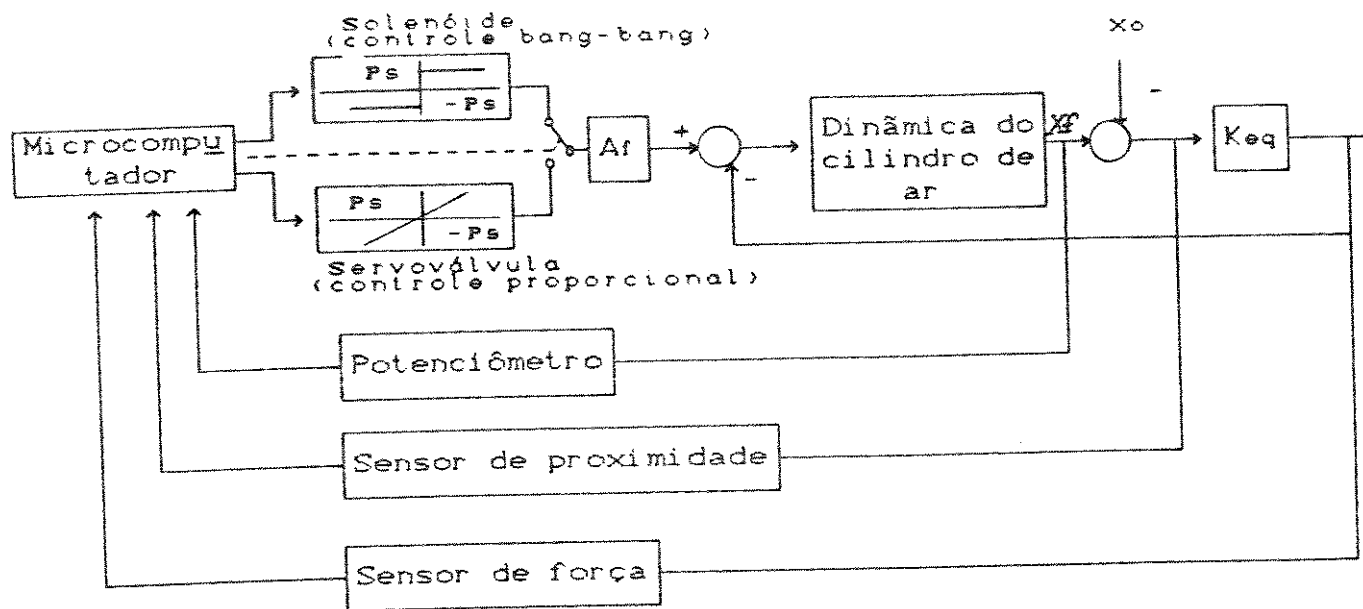


Figura 3.5 - Sistema de controle bimodal de uma garra mecânica com acionamento pneumático. [Parker & Paul 1987]

onde:

A_p : área dos pistão;

K_{eq} : Coeficiente de rigidez do meio;

x_f : Posição do dedo;

x_o : Posição do objeto.

Em [Vossoughi & Donath 1988] é feito uma análise do controle de rigidez do dedo de uma garra mecânica na presença de não linearidades, tais como: carga e velocidades dependentes do atrito e amortecimentos não lineares. No caso, essas não linearidades eram devido ao guia do cabo de aço que realizava a transmissão de movimento. Nesse guia o atrito e o amortecimento dependiam também da posição do cabo.

O estudo desse sistema é dividido em subsistemas linear e não linear. No projeto do controlador de rigidez é usado a técnica de alocação de polos e os parâmetros do modelo não linear são usados para implementar um estimador em tempo real para compensação dos efeitos

não lineares. O esquema geral do controlador e compensador é mostrado na figura 3.6.

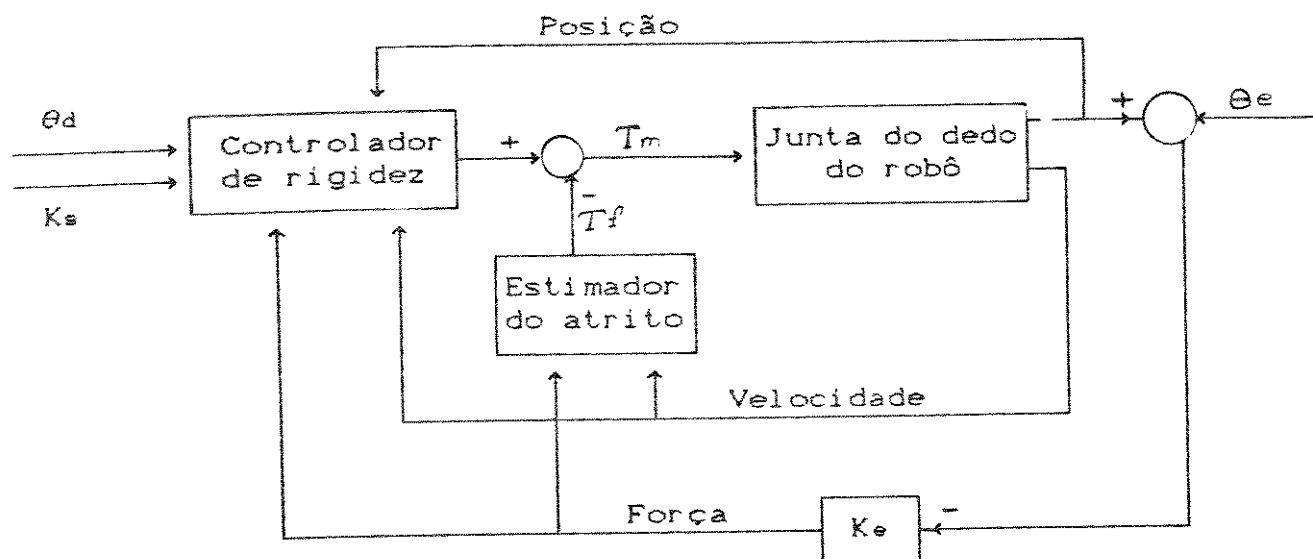


Figura 3.6 - Esquema do compensador e controle de rigidez
[Vossoughi & Donath 1988]

onde:

- θ_d : Posição desejada;
- θ_e : Posição do objeto;
- K_s : Coeficiente de rigidez desejado;
- K_e : Coeficiente de rigidez do meio;
- T_f : Torque de perturbação estimado;
- T_m : Torque motor.

Como foi visto, o assunto de controle de robôs em tarefas de contatos data de um certo tempo, mas que ainda continua em constante estudo. No caso mais específico das garras mecânicas, um dos controladores mais importante é o de rigidez. Desta forma, o objetivo da próxima seção é descrever o controle de rigidez aplicado no controle de garras mecânicas.

Inicialmente são dados os conceitos básicos do controle de rigidez para uma junta genérica de manipulador e depois estes

conceitos são extendidos levando-se em consideração as características específicas do estudo de garras, onde são utilizados conceitos do capítulo 2, como a transformação "Grip".

3.3 - Controle de Rigidez

Como foi visto no capítulo 1, rigidez refere-se a taxa na qual forças e torques na mão aumentam quando está deflexionada de sua posição nominal. O princípio básico do controle de rigidez vem da equação de uma mola sujeita a uma deformação numa dada direção x : (lei de Hooke)

$$f = K \cdot \delta x \quad (3.2)$$

Extendendo essa equação para sistemas elásticos, tem-se:

$$\underline{f} = K_c \cdot \underline{\delta x} \quad (3.3)$$

onde:

$\underline{f}$ = forças generalizadas;

$K_c = \text{diag}(k_x, k_y, k_z, g_x, g_y, g_z)$ matriz que especifica a rigidez do sistema no espaço cartesiano (três translacionais e três rotacionais);

$\underline{\delta x}$ = deslocamento generalizado.

Para transformar uma rigidez desejada em coordenadas cartesianas para coordenadas de juntas são usadas as seguintes relações:

$$\underline{\tau} = J^T \cdot \underline{f} \quad (3.4)$$

$$\underline{\delta x} = J \cdot \underline{\delta \theta} \quad (3.5)$$

onde:

$\underline{\tau}$ = vetor torque das juntas;

J = Jacobiano;

$\delta \underline{x}$ = deslocamento infinitesimal generalizado;

$\delta \theta$ = deslocamento da junta.

Então combinando as equações (3.3), (3.4) e (3.5) obtém-se a seguinte relação:

$$\underline{\tau} = J^T \cdot K_c \cdot J \cdot \delta \theta = K_\theta \cdot \delta \theta \quad (3.6)$$

onde:

$K_\theta = J^T \cdot K_c \cdot J$: matriz de rigidez em coordenadas de juntas.

Pode ser incluído nesta expressão uma determinada força inicial que se deseja imprimir no objeto, seja para suportar a força da gravidade ou outra força de perturbação. Se $\underline{F}_0$ é esta força polarizada no espaço cartesiano, a relação (3.6) passará a ter a seguinte forma:

$$\underline{\tau} = K_\theta \cdot \delta \theta + J^T \cdot \underline{F}_0 \quad (3.7)$$

Um diagrama de blocos de um controlador de rigidez utilizando a relação da equação (3.7) é apresentado na figura 3.7. Nesta figura os blocos no interior do tracejado representam a forma de simular o contato existente entre o manipulador e o meio. Nesse diagrama K_θ representa o coeficiente de rigidez do manipulador, do objeto e do meio.

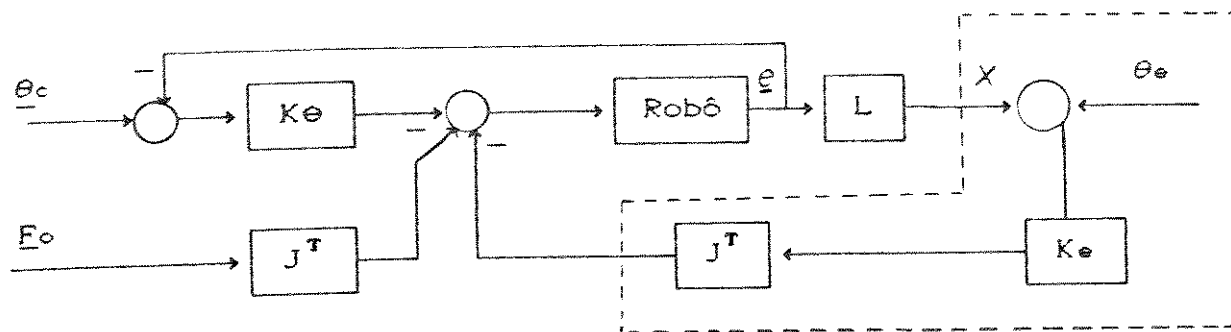


Figura 3.7- Controle de rigidez sem realimentação de força.

Uma outra relação do controle de rigidez pode ser encontrado em [Whitney 1985], onde é realizado uma realimentação de força e uma de velocidade, realimentações estas que poderão melhorar o desempenho da dinâmica do sistema. A relação proposta para uma junta isolada é a seguinte:

$$\tau = K_p.(\theta_o - \theta_l - f_e/K_f) + K_v.\dot{\theta}_l \quad (3.8)$$

onde:

- K_p : Ganho de posição;
- K_v : Ganho de velocidade;
- $1/K_f$: Ganho de esforço;
- θ_o : Posição de referência;
- θ_l : Posição medida;
- $\dot{\theta}_l$: Velocidade medida;
- f_e : Força medida.

O diagrama de blocos do controle de rigidez aplicando esta relação pode ser vista na figura 3.8.

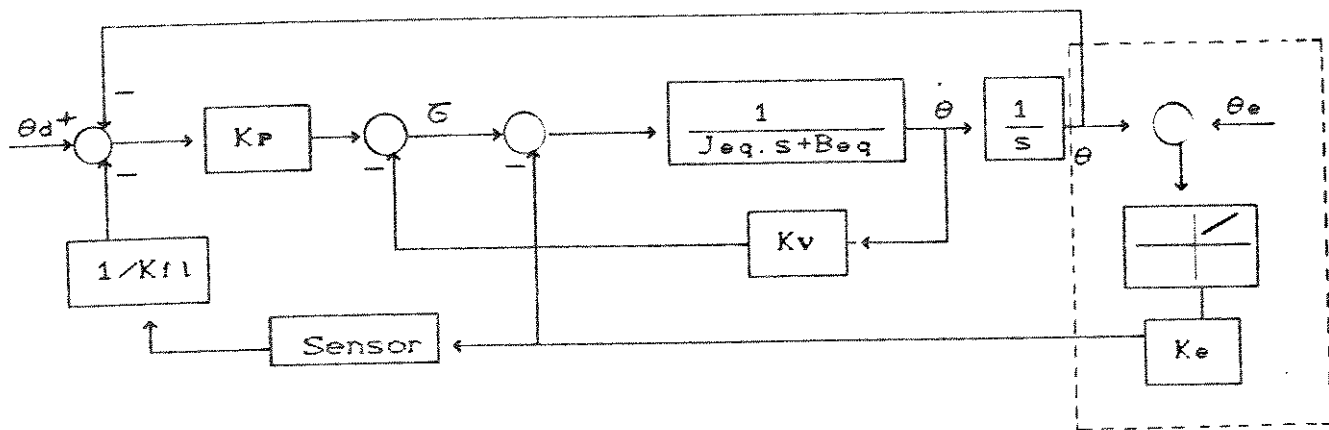


Figura 3.8 - Controle de rigidez com realimentação de força e velocidade. [Whitney 1985]

onde:

θ_d : posição desejada;

θ_e : posição do objeto;

θ : posição da junta;

$\dot{\theta}$: velocidade da junta;

J_{eq} : momento de inércia equivalente da junta;

B_{eq} : Coeficiente de atrito viscoso equivalente da junta;

K_e : coeficiente de rigidez do meio;

A rigidez apresentada pela junta neste caso é dada por:

$$K = \frac{K_P \cdot K_f}{K_P + K_f} \quad (3.9)$$

Também é apresentado em [Whitney 1985] um esquema para controle de rigidez cartesiano. O diagrama de blocos deste esquema está apresentado na figura 3.9.

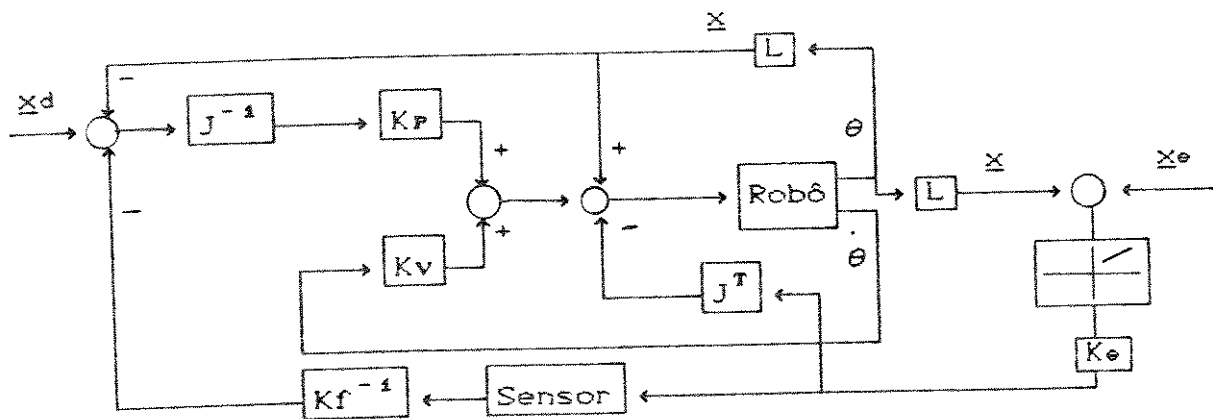


Figura 3.9 - Diagrama de blocos do controle de rigidez cartesiano
[Whitney 1985]

onde:

- $\underline{x}_d$: Vetor de posição de referência;
- $\underline{x}$: Vetor de posições medidas;
- $\underline{x}_e$: Vetor de posição do meio;
- $\underline{\theta}$: Vetor de posições medidas das juntas;
- $\dot{\underline{\theta}}$: Vetor de velocidades medidas das juntas;
- $\underline{K}_P$: Matriz do ganho de posição;
- $\underline{K}_V$: Matriz do ganho de velocidade;
- $\underline{K}_f^{-1}$: Matriz do ganho de força.

Neste caso, a rigidez apresentada pelo robô é dado por:

$$\underline{K} = ((\underline{J}^{-T} \cdot \underline{K}_P \cdot \underline{J}^{-1})^{-1} + \underline{K}_f^{-1})^{-1} \quad (3.10)$$

Observa-se pela expressão (3.10) que a rigidez do robô depende, de forma bastante complexa, de $\underline{K}_P$ e de $\underline{K}_f$.

Em [Salisbury 1981 e 1982] o controle de rigidez é estendido para uma garra articulada, que é o objeto de estudo deste trabalho. Neste caso a matriz de rigidez deve ser aumentada quanto ao número de forças internas que dará o efeito de uma rigidez interna e que pode ser usada para estabilizar uma operação de preensão.

Então a matriz de rigidez terá a seguinte forma:

$$K_c = \text{diag}(K_x, k_y, k_z, g_x, g_y, g_z, k_1, k_2, \dots, k_p)$$

O jacobiano será composto dos jacobianos individuais de cada dedo de tal forma que:

$$J = \begin{bmatrix} \underline{J_1} & \underline{0} & \underline{0} & \underline{0} \\ \underline{0} & \underline{J_2} & \underline{0} & \underline{0} \\ \underline{0} & \underline{0} & \underline{J_3} & \underline{0} \\ \dots & \dots & \dots & \dots \\ \underline{0} & \underline{0} & \underline{0} & \underline{J_n} \end{bmatrix}$$

onde cada J_i relaciona a velocidade da i -ésima junta com as velocidades dos pontos onde existe o contato com o objeto preso.

Então o torque das juntas dos dedos necessário para aplicar uma força generalizada $\underline{\mathcal{F}}$ será dado por:

$$\underline{\tau} = J^T \cdot G^T \cdot \underline{\mathcal{F}} \quad (3.11)$$

e:

$$\underline{\mathcal{F}} = K_c \cdot G \cdot J \cdot \delta \underline{\theta} \quad (3.12)$$

Então, combinando (3.11) e (3.12), obtém-se:

$$\underline{\tau} = J^T \cdot G^T \cdot K_c \cdot G \cdot J \cdot \delta \underline{\theta} \quad (3.13)$$

Neste caso a matriz de rigidez no espaço de juntas será:

$$K_\theta = J^T \cdot G^T \cdot K_c \cdot G \cdot J$$

Donde, a relação para o controle de rigidez de uma garra articulada incluindo uma força polarizada ($\underline{F}_0$) é dada por,

$$\underline{\tau} = K\theta.\delta\theta + J^T.G^T.F_0$$

(3.14)

Foram apresentados vários métodos propostos para o controle de rigidez. A seguir será destacado as vantagens e desvantagens de cada um, que será enumerado de acordo com a figura do diagrama de blocos:

-Figura 3.7 - Este esquema é bastante simplificado, o que facilita sua implementação. No entanto, como não existe uma realimentação de força, será controlado apenas a força em regime permanente, ficando assim, prejudicado o transitório, que é muito importante na operação de preensão;

-Figura 3.8 - Neste esquema, a realimentação de força garante um controle sobre seu transitório. Também existe uma realimentação de velocidade que melhorará a estabilidade do sistema quando a realimentação de força tiver um ganho alto. A desvantagem desse método é a necessidade de várias operações aritméticas, que possivelmente dificultará sua implementação;

-Figura 3.9 - Nos dois primeiros métodos o esquema de controle é feito para apenas uma junta. No caso de um manipulador com mais de um grau de liberdade é necessário a utilização de transformações de coordenadas. Este esquema apresenta o controle de rigidez da figura 3.7 extendido para o controle de um manipulador. Para a garra basta incluir a transformação "Grip". Uma grande dificuldade de implementação deste método é devido a grande quantidade de cálculos aritméticos.

Diante das vantagens e desvantagens apresentadas, e que pretende-se construir uma garra com um grau de liberdade (capítulo 4), será utilizado o esquema de controle de rigidez com realimentações de posição e força, como mostrado na figura 3.10.

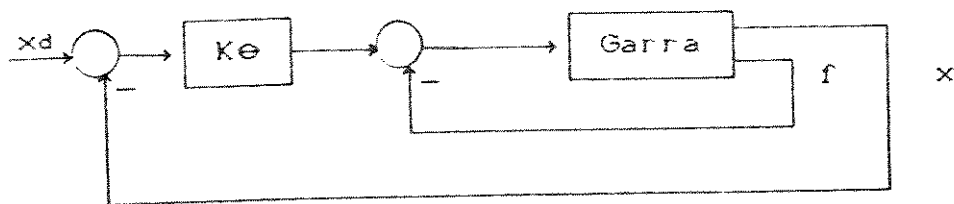


Figura 3.10 - Esquema do controle de rigidez utilizado neste trabalho.

onde:

x_d : posição desejada;
 K_e : constante de rigidez desejada;
 f : força medida;
 x : posição medida.

3.4 - Resumo

Neste foram apresentados, de forma sucinta, os principais tipos de controle de posição e força utilizados em robótica para a execução de tarefas onde existe um contato entre o manipulador e o meio. Dentre estas estratégias de controle, o mais utilizado no controle de garras é o controle de rigidez, por este motivo foram apresentados os princípios básicos que o constituem.

Também foram apresentados vários métodos para implementação do controle de rigidez e com base nas vantagens e desvantagens apresentados por estes, foi escolhido um método para controle de uma garra com um grau de liberdade que será vista no capítulo seguinte.

CAPÍTULO 4

PROJETO E IMPLEMENTAÇÃO DE UMA GARRA MECÂNICA DE DEDO FIXO

4.1 - Introdução

Seguindo os objetivos iniciais deste trabalho será descrito neste capítulo o projeto e implementação de uma garra mecânica de dois dedos sendo um fixo sem nenhuma possibilidade de ação do objeto sobre os seus movimentos.

Inicialmente deve ser considerado que o projeto da garra está inserido em um projeto mais abrangente de um manipulador mecânico com seis graus de liberdade, denominado JECA II, que prevê a utilização de uma garra mecânica com um grau de liberdade, e mais um pulso de dois graus para a sua orientação. Restando então outros três graus para o seu posicionamento no espaço de trabalho pré definido.

Para o controle deste manipulador e a garra projetou-se um Sistema Hierárquico de Controle a Multimicroprocessadores com processamento paralelo [Dias 1991] que pode ser visto na figura 4.1. Além disso está sendo projetado um "Software" para controle de movimentos e trajetórias [Madrid, tese de doutoramento em andamento] que envolve:

- Telas gráficas;
- Definição da linguagem de programação;
- Estruturação do editor;
- Compilador e linker;

- Subrotinas utilitárias;
- Subrotinas de controle;
- Algoritmos de controle de movimentos;
- Controle ponto a ponto;
- Ensino via teclado;
- Algoritmos de interpolação.

Um programa protótipo com algumas funções implementadas encontra-se na Apêndice A

O "hardware" mostrado na figura 4.1 é constituído dos seguintes módulos:

- CPR1 (Controlador Programável de Rôbos 1): É o sistema de controle digital hierárquico distribuído responsável pelo controle do manipulador;
- Módulo Gerador de MLP: Sistema que gera os sinais de controle para o módulo de potência, definindo a largura de pulso de cada junta;
- Módulo de Potência do Recortador: É o circuito responsável pelo controle de potência fornecida pela fonte de alimentação (sistema fonte/bateria);
- Módulo de Sensores: É constituído dos sensores necessários ao controle do robô e os circuitos necessários para tratamento dos sinais obtidos;
- Motores de Corrente Contínua a Imã Permanente.

Desta forma o projeto da garra deve se enquadrar na filosofia proposta do projeto global do sistema. Assim, o sistema da garra terá o esquema básico mostrado na figura 4.2.

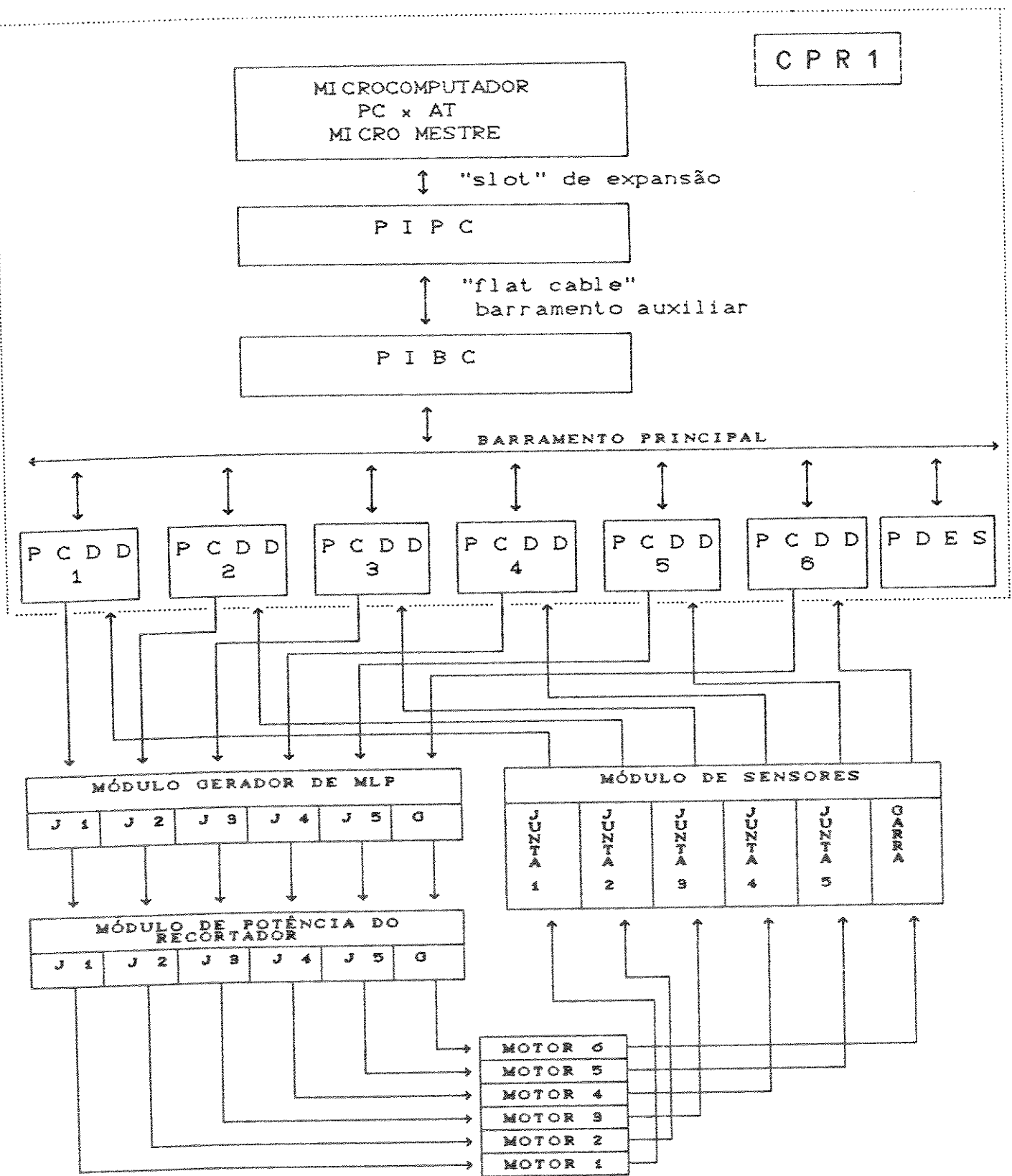


Figura 4.1 - Diagrama de blocos do sistema eletro-eletrônico do projeto JECA II.

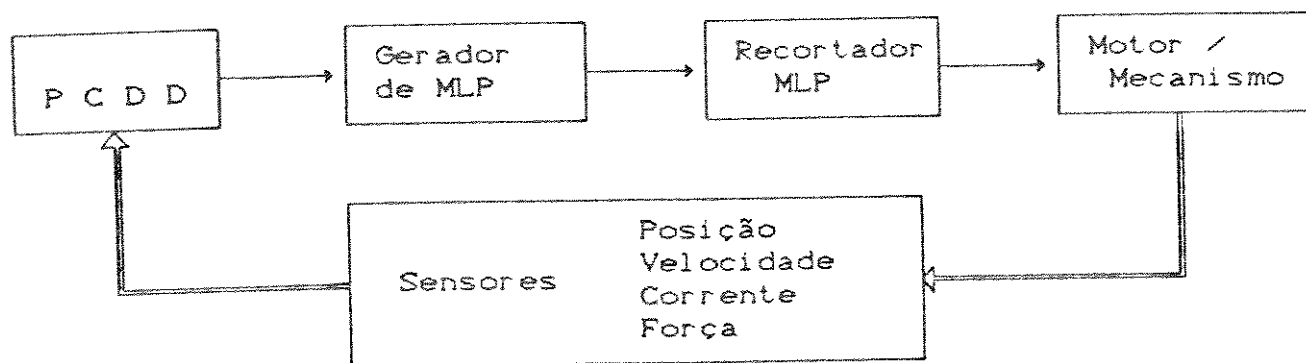


Figura 4.2 - Esquema simplificado do sistema da garra.

Neste capítulo serão descritos o projeto, modelagem e análise de desempenho de cada módulo do esquema mostrado na figura 4.2.

4.2 - Atuador do Tipo Motor cc

O tipo de atuador utilizado no projeto JECA II é o motor de corrente contínua a imã permanente por possuir uma ótima relação torque/velocidade e de relativa facilidade de controle por microcomputadores. Este tipo de motor é largamente utilizado na área de Robótica.

O modelo frequencial de um motor c.c. pode ser representado no diagrama de blocos da figura 4.3 [Eletro-Craft 1973].

Como pode ser observado, a velocidade do motor cc é proporcional à tensão de armadura. No entanto, como foi visto no capítulo 3, o modelo do atuador utilizado no controle de rigidez tem como sinal de entrada um torque. Assim, para que isso ocorra em um motor cc, deve ser utilizado um amplificador de transcondutância, ou seja um amplificador que tem como saída uma corrente que é proporcional a uma tensão de entrada. Isso porque a corrente de armadura e o torque do motor tem uma relação de proporcionalidade:

$$\tau_m = K_t \cdot i$$

(4.1)

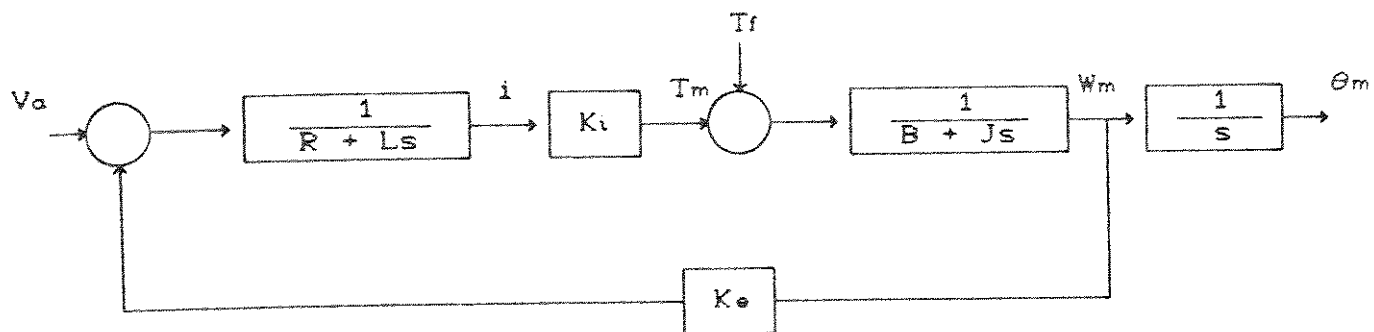


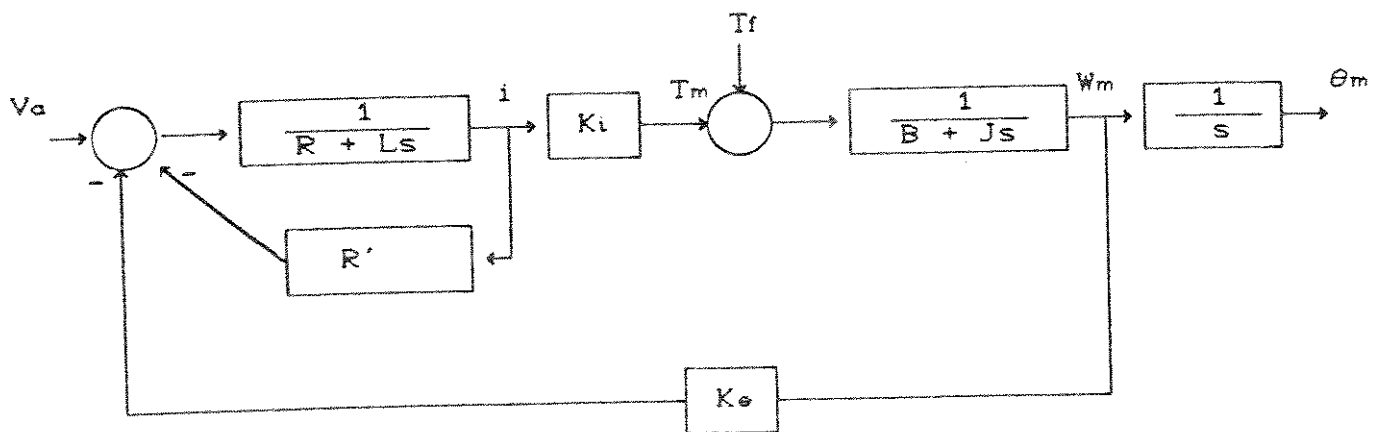
Figura 4.3 - Diagrama de blocos de um motor cc a imã permanente.

onde:

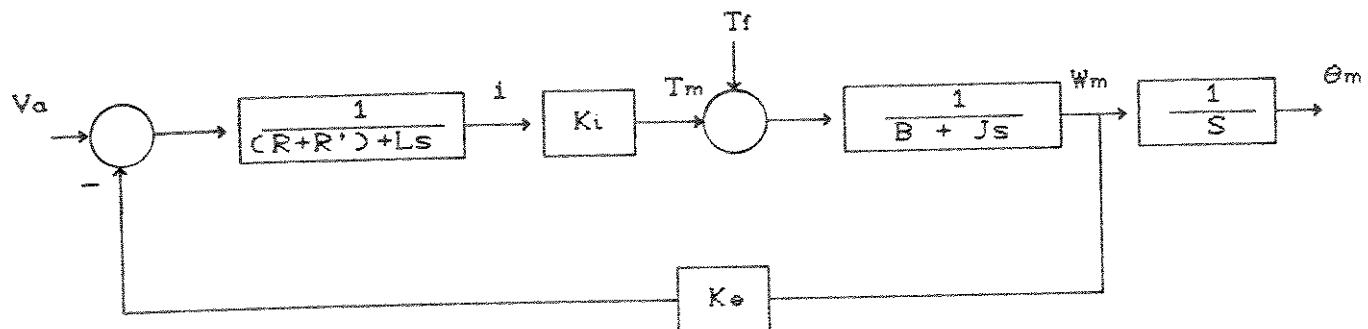
V_a : Tensão de armadura;
 R : Resistência de armadura;
 L : Indutância de armadura;
 K_t : Constante de torque;
 K_e : Constante da força contra-eletromotriz;
 B : Coeficiente de atrito viscoso;
 J : Momento de inércia;
 T_f : Torque de atrito seco.

Uma outra forma de se obter uma corrente de armadura proporcional a tensão de entrada é fazer uma realimentação de corrente, como mostrado na figura 4.4.a e 4.4.b.

R' é o ganho da realimentação de corrente de armadura e é feito suficientemente grande para que a constante de tempo elétrica do motor seja desprezível em relação à constante de tempo mecânica, sendo possível uma resposta elétrica instantânea. Se isto for feito o diagrama de blocos será o da figura 4.5.a que poderá ser simplificado como mostrado na figura 4.5.b.

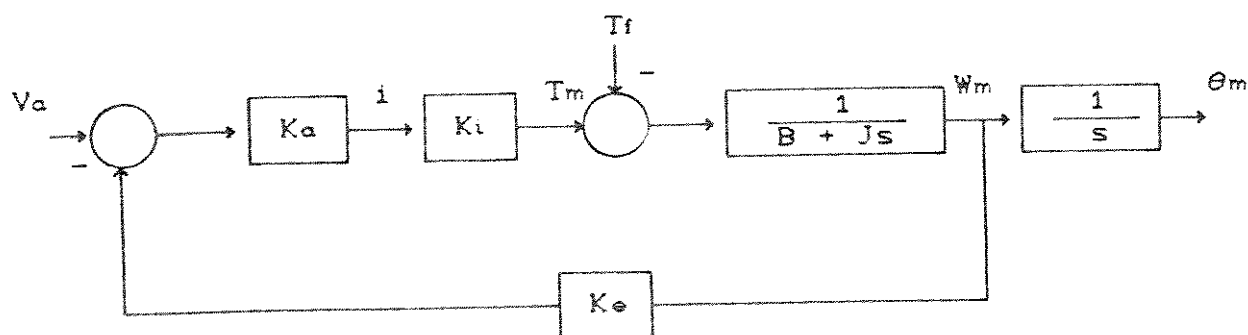


(a)

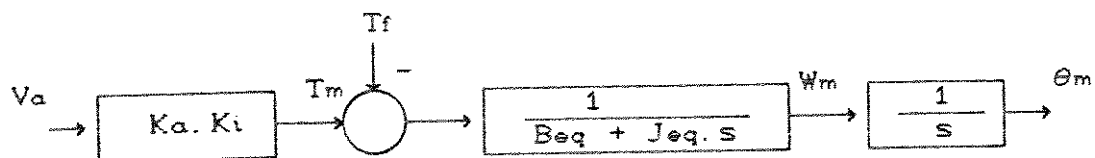


(b)

Figura 4.4 a) Diagrama de blocos de um motor cc com realimentação de corrente;
b) Diagrama de blocos simplificado do item a)



(a)



(b)

Figura 4.5 - a) Diagrama de blocos do motor cc com realimentação de corrente;

b) Diagrama de blocos simplificado do item a).

onde:

$$K_a = 1/(R+R');$$

$$J_{eq} = J;$$

$$B_{eq} = B + K_a \cdot K_i \cdot K_e.$$

Se o motor já apresentar uma constante de tempo elétrica desprezível, comparada com a constante de tempo mecânica, não será estritamente necessário a realimentação de corrente, resultando portanto, $K_a = R^{-1}$. No entanto, esta realimentação pode melhorar a dinâmica da resposta de corrente.

O motor cc utilizado foi um motor aproveitado de uma máquina perfuradora de cartões. Utilizando o método temporal de identificação de parâmetros de um motor cc, proposto em [Lord 1977] e descrito no Apêndice B, foram obtidos os seguintes resultados:

$$\begin{aligned}V_a &= 24 \text{ V;} \\R &= 2.05 \text{ } \Omega; \\L &= 5.54 \text{ mH;} \\J &= 1.96.10^{-3} \text{ Kg.m}^2; \\B &= 4.24.10^{-3} \text{ N.m.s/rad;} \\K_t &= 0.28; \\K_t &= K_e; \\T_f &= 0.06 \text{ N.m;} \end{aligned}$$

Então, como a constante de tempo elétrica ($\tau_e = 2.7\text{ms}$) é muito menor que a constante de tempo mecânica ($\tau_m = 51.38\text{ms}$), os parâmetros para o modelo simplificado da figura 4.5.b serão dados por:

$$\begin{aligned}K_a &= 0,49 \text{ momhs;} \\J_{eq} &= 1,96.10^{-3} \text{ Kg.m}^2; \\B_{eq} &= 42,66.10^{-3} \text{ N.m.s/rad.} \end{aligned}$$

4.3 - Sistema Mecânico

4.3.1 - Apresentação geral do projeto

Levando em consideração que este trabalho é o primeiro da área de garras mecânicas desenvolvido nos laboratórios do DSCE, optou-se pela construção de um mecanismo mais simples. Desta forma, foi escolhido um mecanismo de garra mecânica de dois dedos sendo um fixo, nomenclatura esta, sugerida em [Ferraté 1986]. A transmissão de movimento do motor para o mecanismo é feita através de polias, correias dentadas, fuso e cabos de aço.

pode ser visto na figura 4.6.

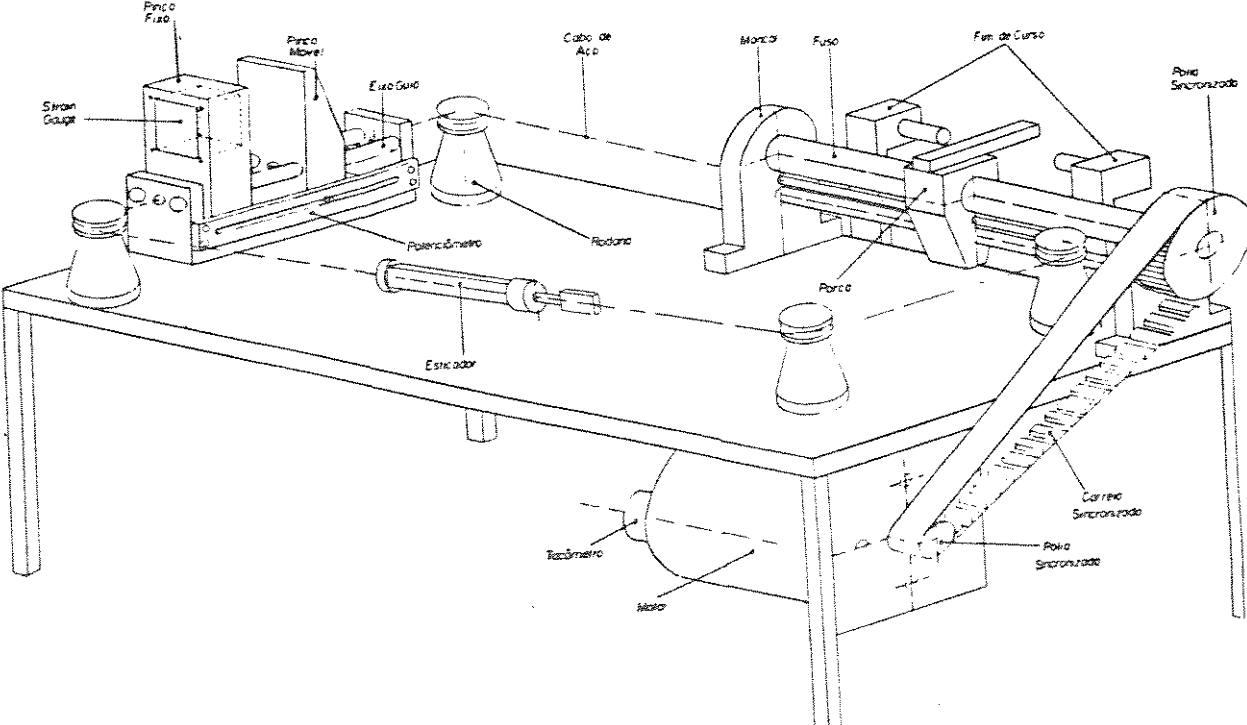


Figura 4.6 - Desenho Mecânico da Plataforma de Teste da Garra Mecânica

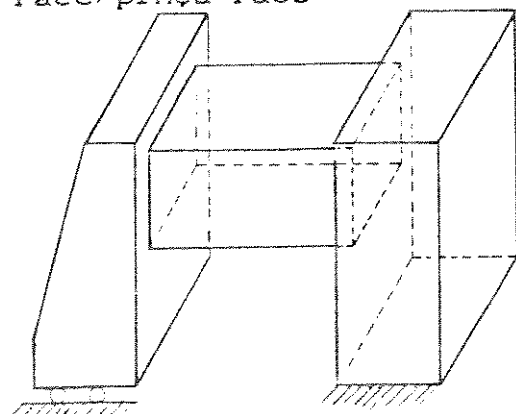
A seguir são feitos estudos e descrições mais detalhados sobre o projeto.

4.3.2 - Análise cinemática e de forças

A análise cinemática estudada no capítulo 2 pode ser aplicada no estudo da garra escolhida, como apresentado em [Salisbury 1982].

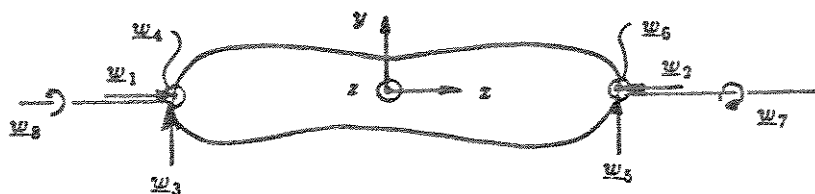
A primeira consideração a ser feita é em relação ao tipo de contato que será escolhido. Se considerar-se que a garra irá prender apenas objetos com superfícies regulares, os contatos poderão ser modelados de acordo com os seguintes casos:

(a) pinça-face/pinça-face



Este tipo de contato poderá ser modelado como planar com atrito ou como toque suave.

Se for do tipo toque suave coincidirá com o exemplo 3 apresentado no capítulo 2:



Assim, a matriz dos "wrenches" principais será dado por:

$$W = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

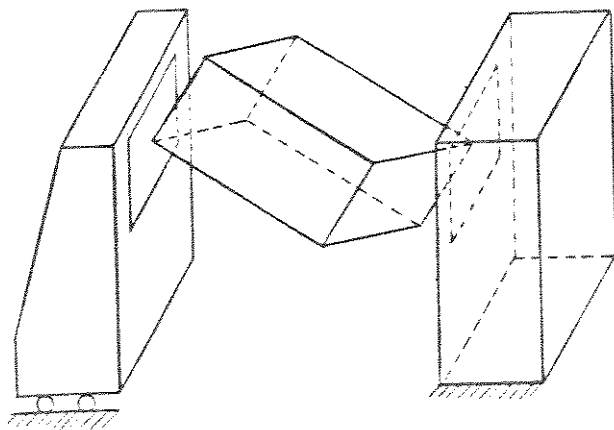
e a solução homogênea será:

$$\underline{c}_h = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \cdot \lambda_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \cdot \lambda_2$$

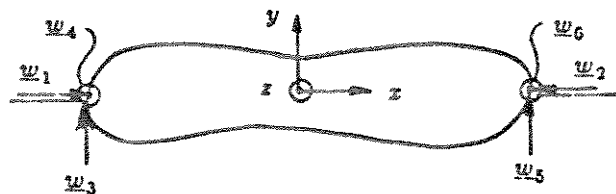
Como foi analisado anteriormente, os dois primeiros "wrenches" devem ser positivos para manter o contato, e observando a solução homogênea $\underline{c}_h$, tem-se que os dois primeiros elementos podem ser feitos arbitrariamente positivos e, portanto, a preensão pode restringir completamente o objeto.

Se o contato for do tipo planar com atrito, irá existir uma matriz W de dimensão 6×12 e a solução homogênea teria dimensão $q \leq n-6$ ou seja, $q \leq 12-6=6$. Isso significa que existirão seis forças internas independentes e a solução homogênea será a combinação linear de seis bases, e não apenas duas como na situação anterior.

b) pinça-vértice/pinça-vértice

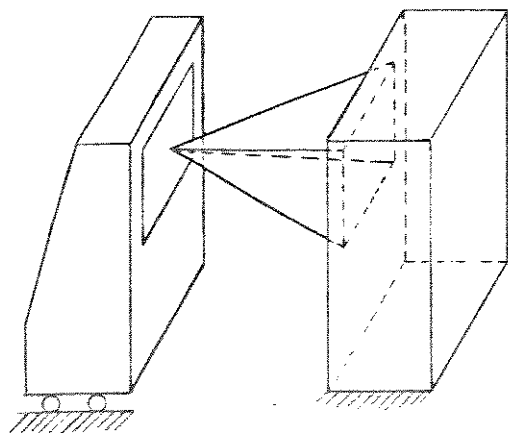


Os contatos, neste caso, devem ser modelados como contatos pontuais com atrito:

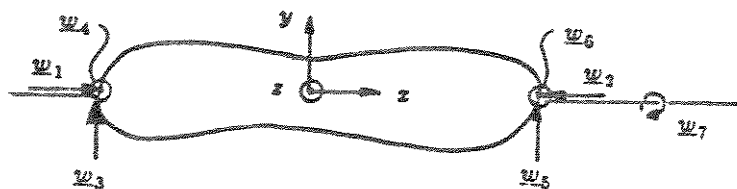


Como foi visto, dois contatos pontuais com atrito não são suficientes para restringir completamente um objeto. Neste caso, existirá uma rotação em torno do eixo x .

c) pinça-vértice/pinça-face



Os contatos podem ser modelados como contato pontual com atrito e contato toque-suave. É o caso do exemplo 5.1 do capítulo 2:



A matriz dos "wrenches" principais será:

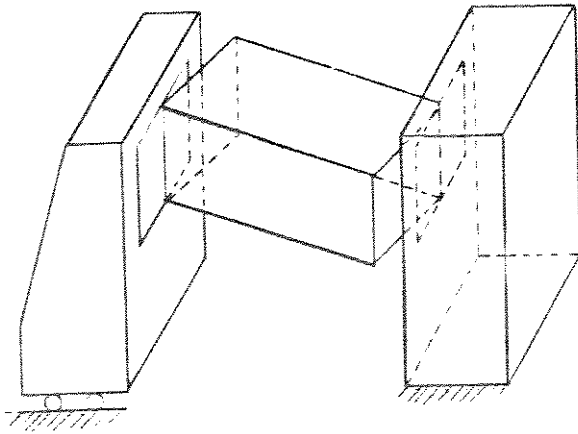
$$W = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Como $n \geq 7$, a solução homogênea terá dimensão $q \leq 1$ e será:

$$\underline{c}_h = [1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0]^T$$

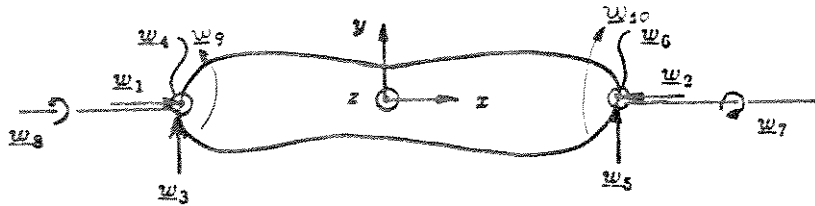
e como os dois primeiros elementos possuem o mesmo sinal, a preensão pode restringir completamente o objeto.

d) pinça-aresta/pinça-aresta



Se o contato for do tipo linear sem atrito, esta preensão não será capaz de restringir completamente um objeto, pois ele possuirá apenas quatro "wrenches" atuando.

Se o contato for do tipo linear com atrito, existirão dez "wrenches" atuando sobre o objeto:



Assim, a matriz dos "wrenches" principais será:

$$W = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

Esta situação é a mesma do caso (a) com toque-suave mais dois "wrenches", assim existirá uma restrição completa do objeto pela preensão. Com $q \leq 4$ existirá uma solução homogênea de dimensão quatro e consequentemente quatro forças internas de preensão.

Em todos estes casos o atrito é considerado ativo e de magnitude suficiente para manter o contato. No entanto, sabe-se que ele depende dos materiais envolvidos (coeficiente de atrito), e da intensidade da força normal ao contato, que nos casos expostos acima são os "wrenches" w_1 e w_2 . Assim, a intensidade desses "wrenches" determinarão se a preensão é estável ou não para uma determinada perturbação.

Dentre estes casos, o (a) é o mais estável, pois nos demais, dependendo do ângulo entre a pinça e o objeto, existirá uma rotação do objeto em relação a pinça. Um estudo deste problema é feito em [Mason 1981]. Portanto, no planejamento da preensão este fato deve ser levado em consideração.

4.3.3 - Descrição e modelagem do sistema motor-transmissão-pinça móvel

O Sistema de transmissão do movimento de rotação do motor para a pinça móvel da garra é constituído por polias, correia dentada, fuso e cabo de aço, conforme se viu na figura 4.6.

A primeira etapa da transmissão é feita por polia e correia dentada, conforme figura 4.7.

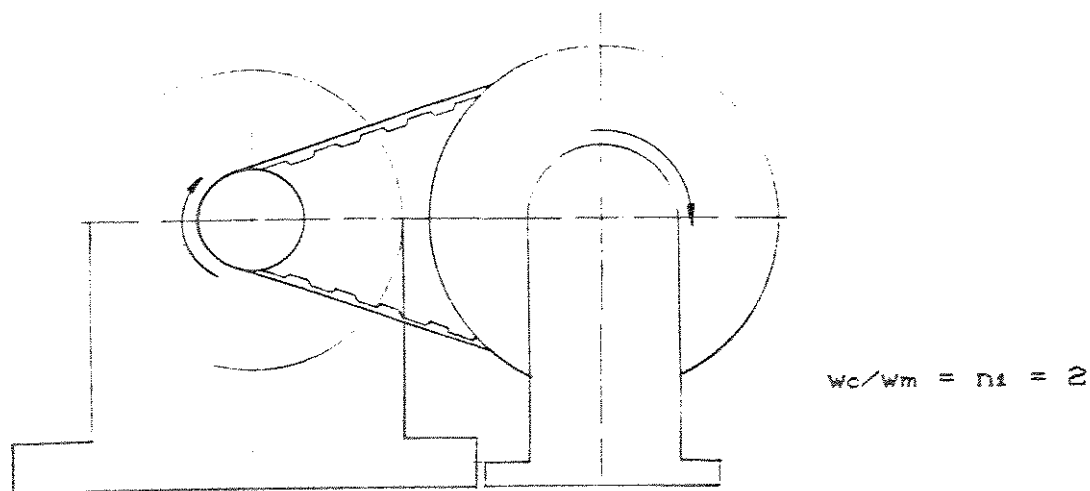


Figura 4.7 - Transmissão polia-correia dentada

Considerando que a correia dentada possui uma certa elasticidade, ela causará um efeito semelhante a torção do eixo do motor. Este efeito pode ser modelado como em [Cabrera 1990], esquematizado no diagrama de blocos da figura 4.8.

Logo após o redutor de velocidade é feita uma conversão de movimento rotacional para movimento translacional através de um fuso e cabos de aço. Essa transmissão está esquematizada na figura 4.9.

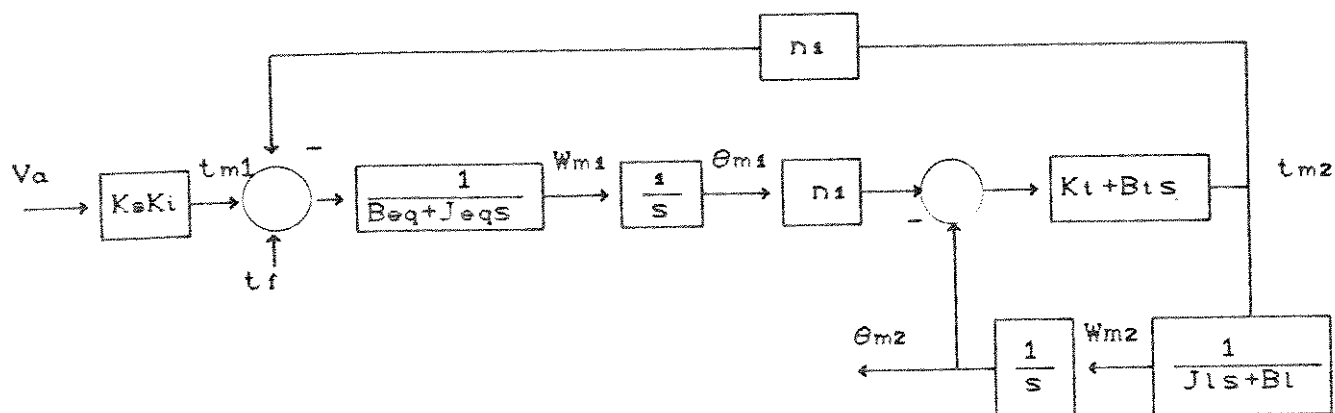


Figura 4.8 - Diagrama de blocos do motor e transmissão polia-correia dentada com torção.

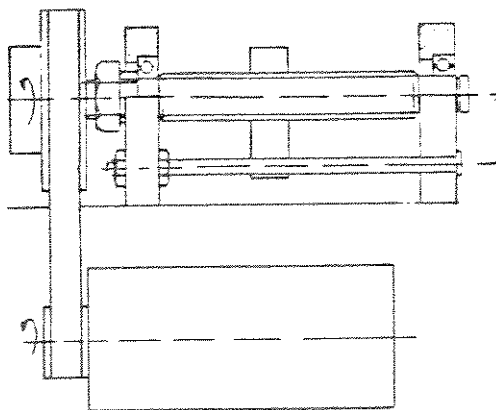


Figura 4.9 - Esquema da transmissão com o fuso

O fuso transfere o movimento rotacional do redutor em movimento translacional - $(T_{m2}, w_{m2}, \theta_{m2}) \rightarrow (F_t, v_t, x_t)$, onde a relação da conversão é :

$$x_t / \theta_{m2} = n_2 = 404,24 \cdot 10^{-3} \text{ m/rad}$$

e portanto,

$$x_1/\theta_{m2} = w_{m2}/v_t = F_t/T_{m2} = n_2 \tag{4.2}$$

O fuso também possui a propriedade de que se o módulo da soma de todas as forças atuantes sobre a pinça, exceto a força de transmissão F_t , for maior que o módulo da força de transmissão, a velocidade v_t será nula. Essa característica pode ser representada por:

$|w_{m2}| - |v_t/n_2| > 0 \rightarrow \text{movimento normal}$

e

$|w_{m2}| - |v_t/n_2| \leq 0 \rightarrow v_t = 0$

Assim, a incorporação do fuso na transmissão pode ser representado pelo diagrama de blocos da fig.4.10.

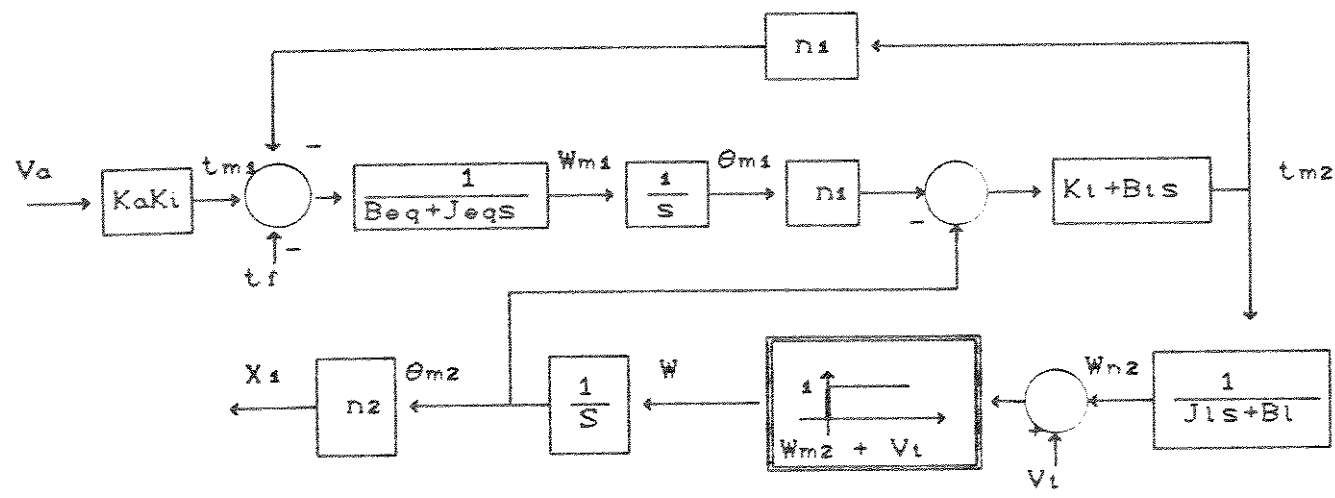


Figura 4.10 - Diagrama de blocos do sistema motor-polia/correia dentada-fuso.

O fuso irá movimentar a massa M_t , que por sua vez transferirá o movimento para a pinça móvel da garra através de um cabo de aço, dirigido adequadamente por polias.

Como o cabo de aço tem uma certa flexibilidade, irá existir uma interação dinâmica entre a massa da transmissão (M_t) e a massa da pinça móvel. Essa interação pode ser modelada por um sistema massa-mola-amortecedor. Um diagrama simplificado pode ser visto na figura 4.11.

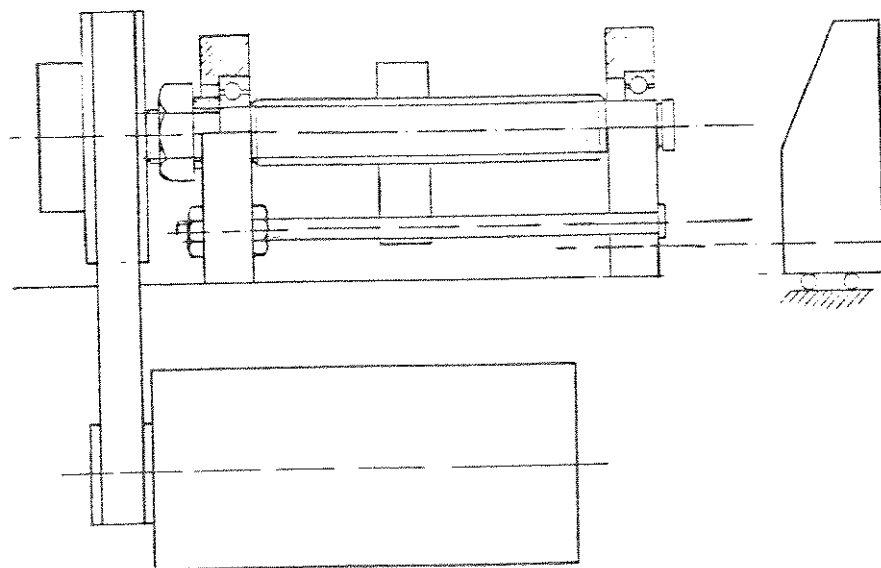
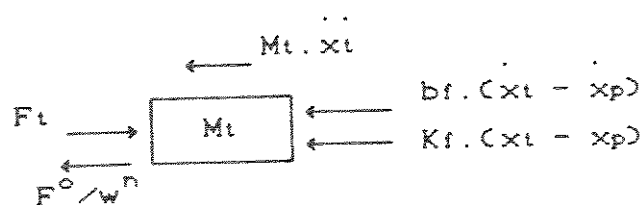


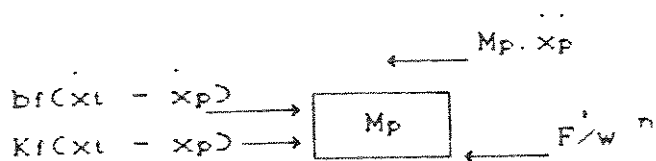
Figura 4.11 - Esquema simplificado do sistema motor-polia/correia-fuso-cabo de aço-pinça.

Fazendo o diagrama de corpo livre e aplicando o princípio de d'Alembert, pode-se obter:

- transmissão - pinça móvel



$$\ddot{x}_t = \frac{1}{M_t} \left(F_t - b f.(\dot{x}_t - \dot{x}_p) - K f.(x_t - x_p) - \frac{F^o}{w^n} \right) \quad (4.3)$$

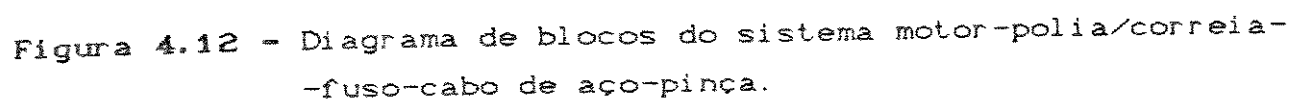


$$\ddot{x}_p = \frac{1}{M_p} (b_f(\dot{x}_t - \dot{x}_p) + K_f(x_t - x_p) - \frac{F^o}{w^n}) \quad (4.4)$$

Onde :

- F_t : força aplicada no sistema de transmissão;
- b_f : coeficiente de atrito viscoso do fio;
- K_f : coeficiente de elasticidade do fio;
- x_t : deslocamento do sistema de transmissão;
- $\dot{x}_t$: velocidade do sistema de transmissão;
- $\ddot{x}_t$: aceleração do sistema de transmissão;
- F^o/w^n : força de atrito seco entre a transmissão e o trilho;
- x_p : deslocamento da pinça.

Então, o sistema completo que representa a garra de preensão de dois dedos sendo um fixo pode ser dado pelo diagrama de blocos da figura 4.12.



A interação entre a garra e o objeto pode se dar em três situações distintas:

- No primeiro caso a pinça móvel irá deslocar-se juntamente com o objeto até entrar em contato com a pinça fixa, quando ocorrerá um impacto.

No segundo caso a pinça móvel irá deslocar-se até entrar em contato com o objeto que já estará em contato com a pinça fixa, ocorrendo também um impacto.

No terceiro caso a pinça móvel irá deslocar-se até entrar em contato com o objeto, ocorrendo um primeiro impacto, depois a pinça móvel e o objeto se deslocarão juntos até entrar em contato com a pinça fixa, ocorrendo um segundo impacto.

Nesses três casos pode-se ter quatro situações de interações dinâmicas:

- Movimento da pinça móvel no espaço livre;
- Contato entre a pinça móvel e o objeto;
- Movimento da pinça móvel-objeto no espaço livre;
- Contato entre a pinça móvel, objeto e a pinça fixa, mais precisamente com a célula de carga.

Na primeira situação existirá interação entre o motor, transmissão e a pinça móvel. Esta interação foi modelada na seção anterior e representada no diagrama de blocos da figura 4.12.

Na segunda situação existirá interação entre o motor, transmissão, pinça móvel e o objeto. A interação até a pinça móvel coincide com a situação anterior e a interação entre a pinça móvel e o objeto pode ser esquematizada como na figura 4.13.

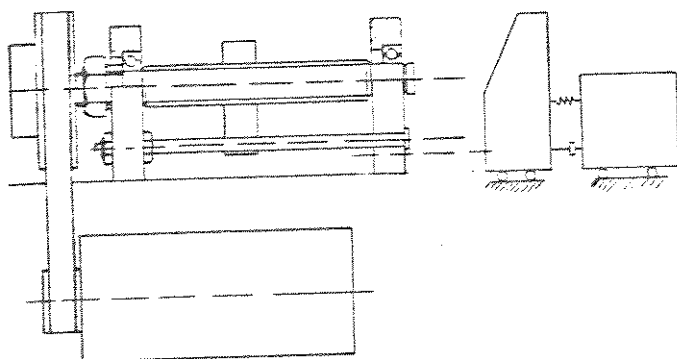
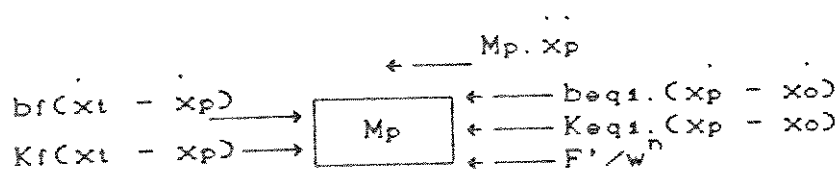


Figura 4.13 - Interação entre a pinça-móvel e o objeto.

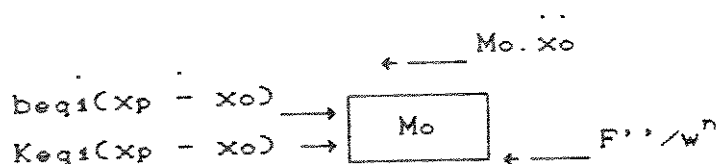
Fazendo o diagrama de corpo livre para a pinça móvel e o objeto, obtém-se:

- pinça móvel - objeto



$$\ddot{x}_p = \frac{1}{M_p} (b_f(\dot{x}_i - \dot{x}_p) + K_f(x_i - x_p) - b_{eq1}(\dot{x}_p - \dot{x}_o) - K_{eq1}(x_p - x_o) - \frac{F'}{w^n}) \quad (4.5)$$

e



$$\ddot{x}_o = \frac{1}{M_o} (b_{eq1}(\dot{x}_p - \dot{x}_o) - K_{eq1}(x_p - x_o) - \frac{F''}{w^n}) \quad (4.6)$$

onde:

$$K_{eq1} = \frac{K_p \cdot K_o}{K_p + K_o} ;$$

$$b_{eq1} = \frac{b_p \cdot b_o}{b_p + b_o} ;$$

K_p = Coeficiente de elasticidade da pinça;

K_o = Coeficiente de elasticidade do objeto;

b_p = Coeficiente de atrito viscoso da pinça;

b_o = Coeficiente de atrito viscoso do objeto;

Assim a segunda situação pode ser modelada com as equações (4.5) e (4.6) juntamente com o sistema motor-transmissão. Este modelo está representado pelo diagrama de blocos da figura 4.15.

A terceira situação também é modelada pelas equações (4.5) e (4.6) e o sistema motor-transmissão.

Na quarta situação existirá interação entre a pinça móvel, o objeto e a pinça fixa. Esta situação está esquematizada na figura 4.14.

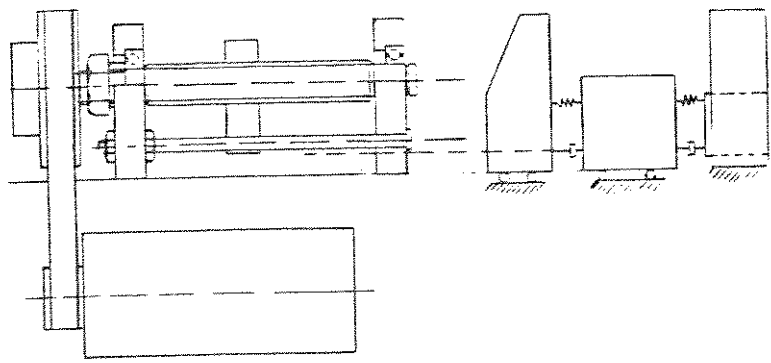
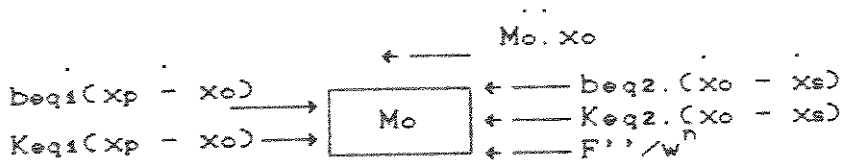


Figura 4.14 - Interação entre pinça móvel-objeto-pinça fixa

Fazendo o diagrama de corpo livre do objeto e da pinça fixa, obtém-se:

- pinça móvel - objeto - pinça fixa



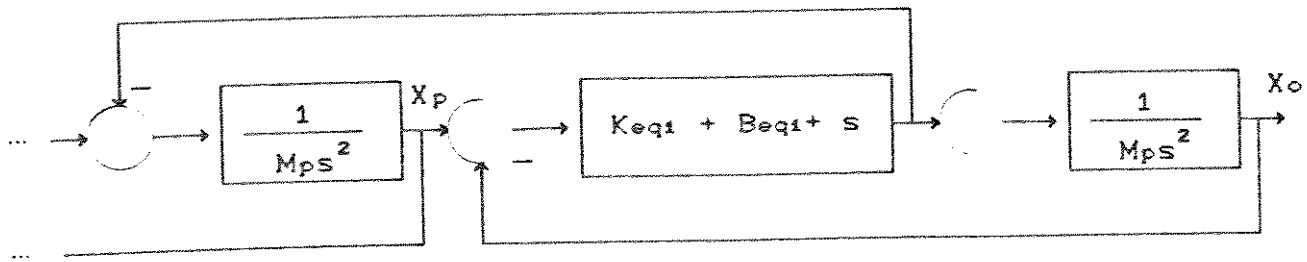
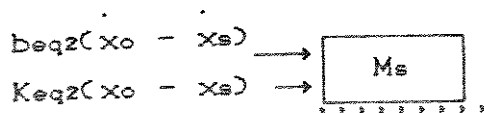


Figura 4.15 - Diagrama de blocos do sistema -pinça móvel-objeto

$$\ddot{x}_o = \frac{1}{M_o} (b_{eq1}(\dot{x}_p - \dot{x}_o) + K_{eq1}(x_p - x_o) - b_{eq2}(\dot{x}_o - \dot{x}_s) - K_{eq2}(x_o - x_s) - F''/w^n) \quad (4.7)$$

e



$$b_{eq2}(\dot{x}_o - \dot{x}_s) - K_{eq2}(x_o - x_s) = 0 \quad (4.8)$$

onde:

$$K_{eq2} = \frac{K_s \cdot K_o}{K_s + K_o} ;$$

$$b_{eq2} = \frac{b_s \cdot b_o}{b_s + b_o} ;$$

K_s = Coeficiente de rigidez da célula de carga;

K_o = Coeficiente de rigidez do objeto;

b_s = Coeficiente de atrito viscoso da célula de carga;

b_o = Coeficiente de atrito viscoso do objeto;

Assim a interação entre a pinça móvel, objeto e a pinça fixa é modelada pelas equações (4.7) e (4.8) e pelo sistema motor-transmissão-pinça móvel. Esse modelo está representado pelo diagrama de blocos da figura 4.16.

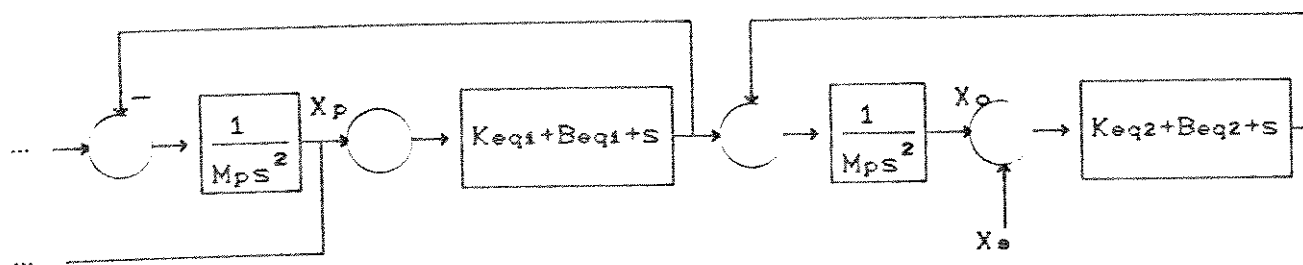


Figura 4.16 - Diagrama de blocos do sistema motor-transmissão-pinça móvel-objeto-pinça fixa.

4.4 - Acionamento: Recortador por MLP em quatro quadrantes

O motor cc é acionado por um recortador a transistores bipolares em configuração de ponte H, podendo assim trabalhar em quatro quadrantes. Esse recortador utiliza a técnica de modulação por largura de pulso para obter uma tensão média de saída proporcional ao sinal de entrada, e foi escolhido porque apresenta vantagens sobre o acionamento linear, tais como: [Palhares 1988]

- Diminui o atrito seco do motor cc, não linearidade do tipo zona morta;
- Pode compensar folgas em engrenagens, não linearidade do tipo histerese;
- A configuração do recortador em ponte H propicia a regeneração de energia, fator importante quando a fonte de alimentação são baterias.
- Proporciona um melhor rendimento de energia, pois ele trabalha somente no corte ou saturação dos transistores e isso leva a uma menor perda de energia.

A grande desvantagem é a ondulação causada pelo chaveamento, que levam a oscilações subharmônicas ou caóticas, dependendo da região de operação que ocorrer durante a execução de uma determinada tarefa, sendo que esta região depende de vários fatores tais como frequência de chaveamento, ganho do sistema, tipo de algoritmo de controle e variações na carga que podem levar a uma mudança do ponto de operação, fato que é muito comum em juntas robóticas, devido às variações na geometria do braço que faz com que os motores de cada junta vejam sempre um momento de inércia variando continuamente no tempo.

A forma de onda sobre o motor neste tipo de recortador está mostrada na figura 4.17.

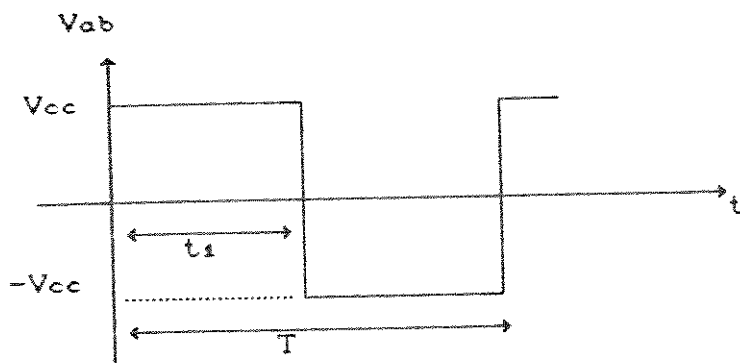


Figura 4.17 - Forma de onda da tensão sobre um motor acionado por um recortador MLP de quatro quadrantes.

A frequência ($1/T$) deve permanecer fixa e a variação da tensão média ser dada pela variação do ângulo de comutação, largura de pulso :

$$\theta = (t_1/T)$$

(4.9)

Esta freqüência de chaveamento deve ser pelo menos duas vezes maior do que a freqüência mecânica do motor e é limitada somente pelo dispositivo de chaveamento sendo que para transistores bipolares pode-se chegar na prática pelo menos dez vezes maior. O motor em questão possui constante de tempo mecânica de 51.8ms. Assim, a freqüência de chaveamento será escolhida para operar em aproximadamente 200 Hz ou mais se os transistores utilizados permitirem.

Em sistemas chaveados todas análises podem ser feitas através de valores médios e do conceito de ganho equivalente, no entanto é necessário uma análise local para garantir a estabilidade local. Uma forma de se fazer essa análise é a por modelos recorrentes, que é bastante estudado em [Palhares 1988], [Araújo 1988] e [Álvarez 1991].

Supondo que se possa utilizar o conceito de valor médio, a tensão média sobre o motor será dada por:

$$V_m = 48.(\theta - 1/2) \quad (4.10)$$

Por esta expressão tem-se que para:

$$\theta < 1/2 \implies -24 \leq V_m < 0 \text{ V}$$

$$\theta = 1/2 \implies V_m = 0 \text{ V}$$

$$\theta > 1/2 \implies 0 < V_m \leq 24$$

O diagrama de blocos do recortador utilizado está apresentado na figura 4.18.

Nesse diagrama pode ser observado que existe um módulo que gera o sinal de modulação MLP. Esse módulo é detalhado na figura 4.19. Nele existe um contador de oito bits que fica contado ciclicamente, com freqüência dada pelo circuito do relógio. Esta contagem é então comparada, através de um circuito comparador de oito bits, com uma palavra de controle de oito bits do controlador digital.

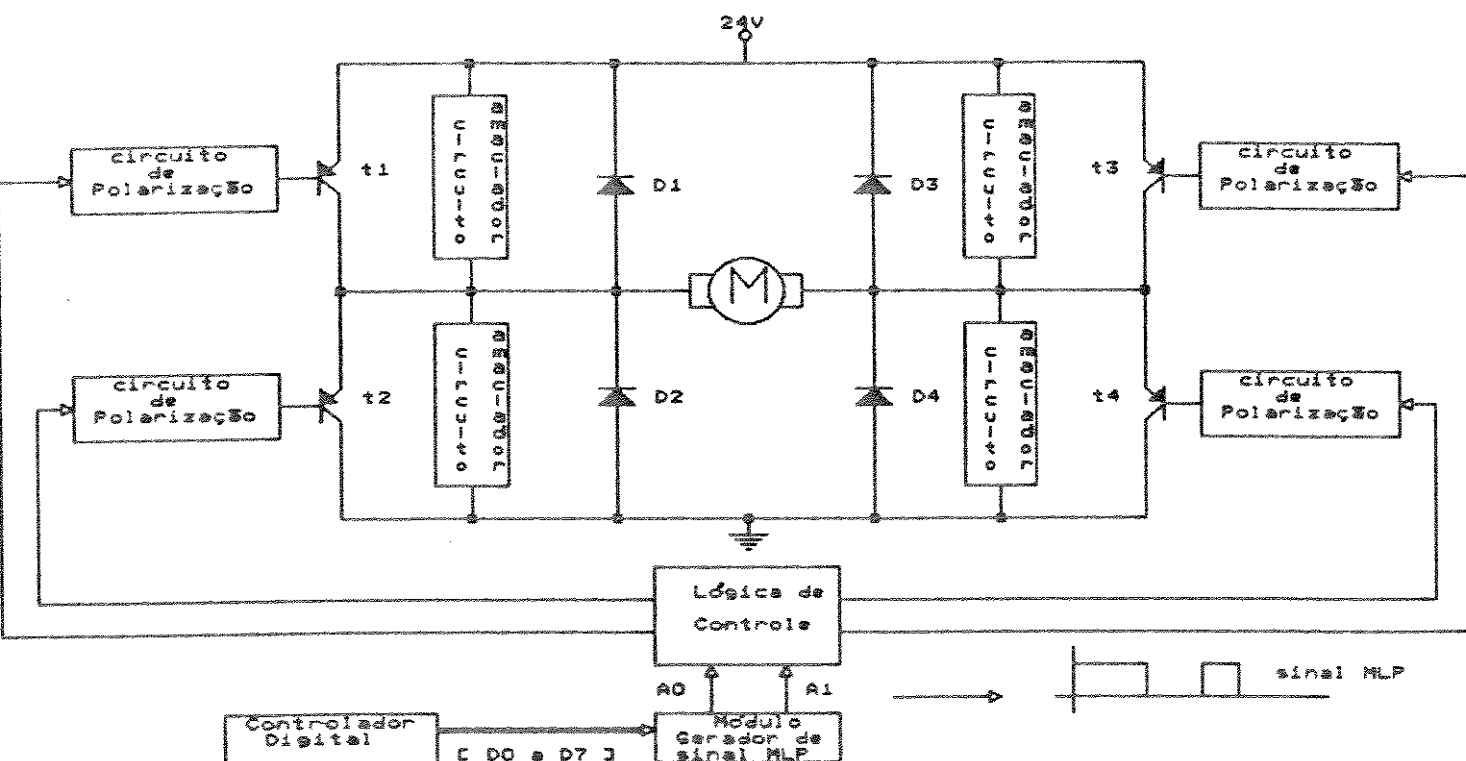


Figura 4.18

- a) Diagrama de blocos do recortador e seu sistema de controle;
- b) Forma de onda da tensão sobre o motor V_{ab} .

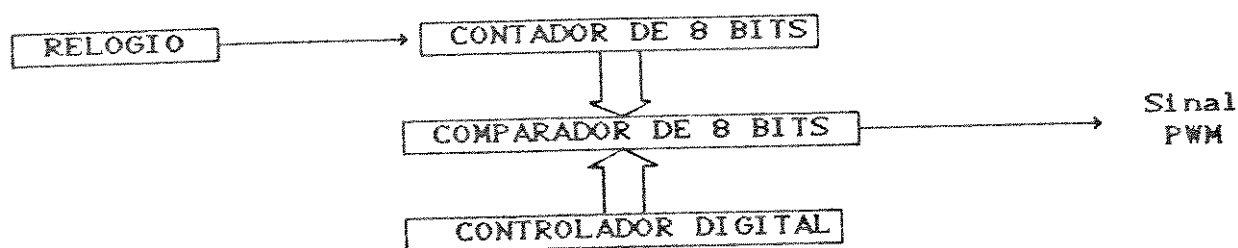


Figura 4.19 - Diagrama de blocos do gerador de PWM

Assim, a saída deste circuito será:

- 5V se a palavra do contador $\leq$ palavra do controlador
- 0V se a palavra do contador $\geq$ palavra do controlador

Como estão sendo utilizados oito bits, existem 256 possibilidades de ângulo de comutação. A relação entre a palavra de controle vinda do controlador e o ângulo de comutação é dado na tabela 4.1.

Tabela 4.1 - Relação entre a palavra de controle e o ângulo de comutação.

Palavra de controle	Ângulo de comutação
00h	1
01h	255/256
02h	254/256
.	.
.	.
.	.
80h	1/2
.	.
.	.
.	.
FDh	3/256
FEh	2/256
FFh	1/256

O sinal MLP gerado segue para o bloco da lógica de controle. Este bloco possui duas entradas, A_0 e A_1 . Esses sinais determinam se o recortador trabalha em dois ou quatro quadrantes e isso é feito segundo a seguinte combinação: se $A_0 = 1$ e $A_1 = \text{sinal MLP}$, o recortador funcionará em dois quadrantes; se $A_0 = \text{sinal MLP}$ e $A_1 = 1$, o recortador funcionará em quatro quadrantes. No caso da garra o recortador funcionará sempre em quatro quadrantes.

Além de selecionar a configuração de dois ou quatro quadrantes, a lógica de controle determina quais transistores devem estar cortados e quais saturados para seguir o sinal MLP. Para $V_{ab} = 24 \text{ V}$, T_1 e T_4 devem estar saturados e T_2 e T_3 devem estar cortados. Para $V_{ab} = -24 \text{ V}$, T_1 e T_4 devem estar cortados e T_2 e T_3 devem estar saturados.

A parte de potência é constituída por circuitos amaciadores, dois transistores NPN, dois transistores PNP e quatro diodos. Os transistores formam a ponte H e os diodos são os de livre circulação. Os circuitos amaciadores dissipam parte da energia perdida nos transistores devido ao chaveamento.

O recortador projetado apresenta as seguintes características:

- Frequência de chaveamento: 1 KHz (podendo variar de 200 Hz a 5KHz)
- Corrente máxima de pico: 40 A
- Corrente contínua: 8 A
- Tensão de alimentação: 24 V

4.6 - Controlador Digital

4.6.1 - Descrição geral do CPR1

Como foi visto na introdução, o controle do JECA II é feito pelo controlador programável de robôs CPR1. Esta seção descreverá sucintamente este sistema, em especial a placa controladora de juntas. Detalhes de projeto, esquemas e análise de desempenho podem ser encontrados em [Marcus 1991].

O diagrama de blocos deste sistema está mostrado na figura 4.20.

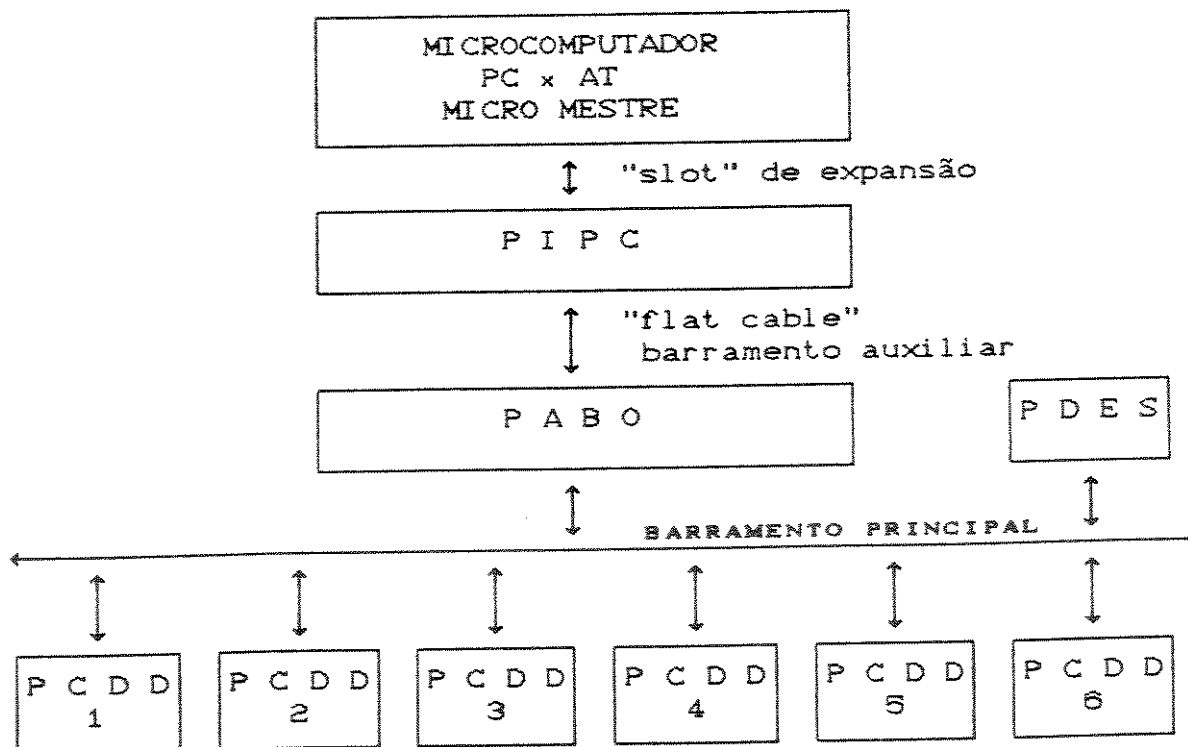


Figura 4.20 - Diagrama de blocos do CPR1

Este sistema é composto dos seguintes módulos:

- Micro Mestre: é um microcomputador responsável pela supervisão das placas controladoras de juntas e pelo controle de trajetórias do manipulador. Este micro é compatível com o IBM-PCxAt e possui a seguinte configuração:

- CPU : 80286
- Coprocessador: 80287
- Memória de usuário: 640 Kbytes
- Memória da placa EGA: 256 Kbytes
- Memória estendida: 384 Kbytes
- Expansão de memória 2 Mbytes
- Saída serial e paralela
- Controladora de disco rígido e flexível
- Controladora de monitor EGA;

- Placa de Interface do PC (PIPC): Tem a função de gerar, receber e coordenar os sinais do barramento auxiliar e conseqüentemente do barramento principal;

- Placa Amplificadora do Barramento (PABO): possui a função de fortalecer os sinais provenientes da PIPC, que são transmitidos por meios de cabos até o barramento principal;

- Placa de Controle Digital de Juntas (PCDD): é um microcontrolador responsável pelo controle de juntas do manipulador;

- Placa Digital de Entrada e Saída (PEDES): monitora o espaço de trabalho e ferramentas do manipulador.

Uma das placas do tipo PCDD está reservada ao controle da garra, assim esta será vista com mais detalhes na subseção seguinte.

4.6.2 - Descrição da PCDD

Esta placa foi projetada para ser dedicada a controle de processos, que neste caso é uma garra mecânica.

A arquitetura geral pode ser vista no diagrama de blocos da figura 4.21.

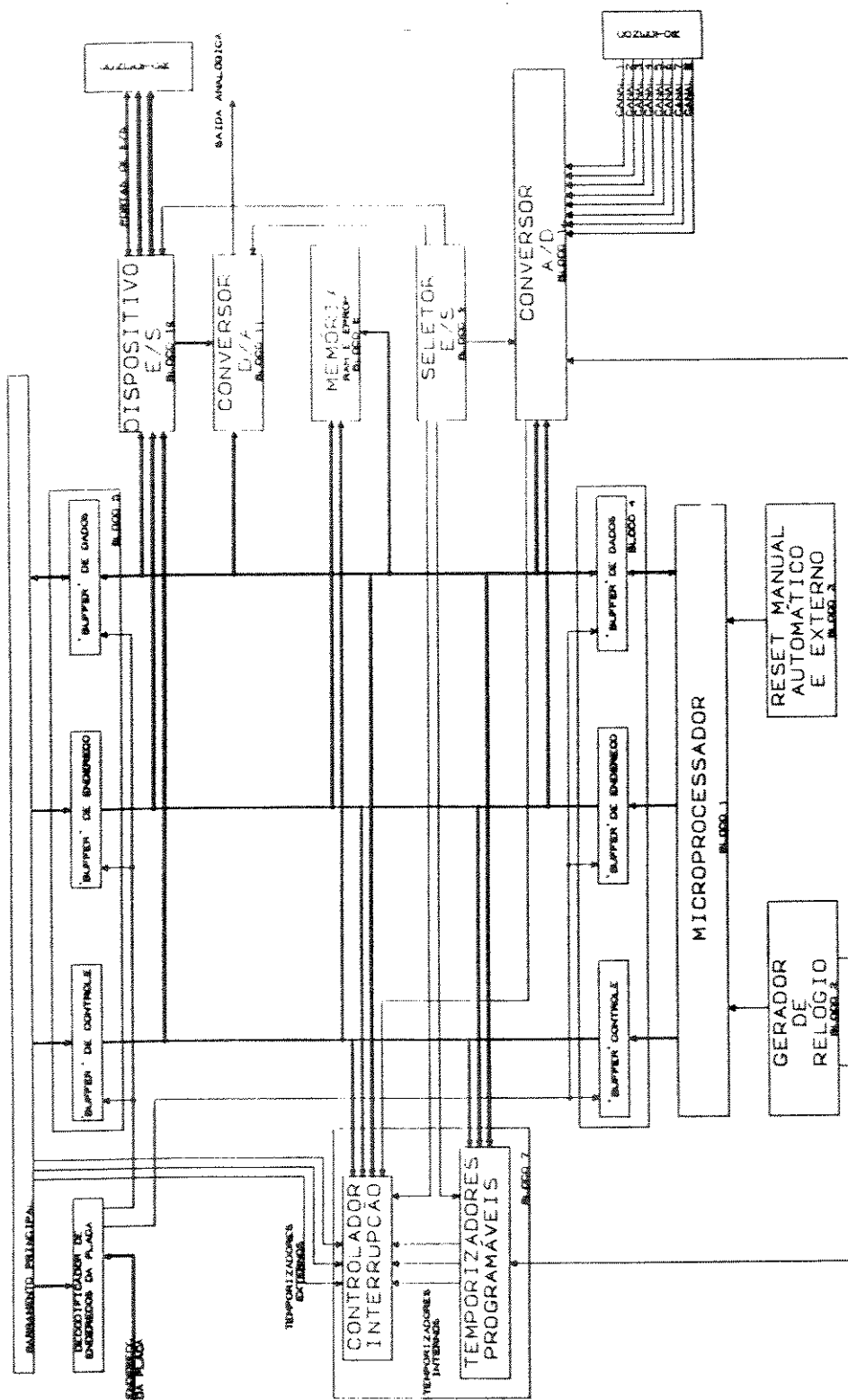


Figura 4.21 - Diagrama de blocos da PCDD

A seguir são descritos, resumidamente, os blocos da figura

4.21:

- Blocos 1, 2, 3, 4 e 8

Formam o circuito básico de um microcomputador. O microprocessador utilizado é o Z80 com uma frequência de relógio de 4MHz.

A memória é dividida em 16Kbytes para EPROM, onde ficará residente as rotinas de inicialização, interrupção, rotinas usadas pelo programa principal (p.e. operações aritméticas) e o programa monitor que é responsável pela coordenação do sistema; e 16 Kbytes para RAM, onde serão as variáveis de programa e tabelas usadas pelo programa principal. O mapa de memória da PCDD é distribuído da seguinte forma:

EPROM	→	0000h a 3FFFh
RAM	→	4000h a 7FFFh

- Blocos 5 e 6

Fazem a interface com o barramento principal, e conseqüentemente com o Micro Mestre.

- Bloco 7

O sistema possui três canais de temporização que são usados para dar sincronismo na execução das rotinas que possuem um tempo determinado para execução. Este circuito é alimentado por uma frequência de relógio de 500 KHz, o que possibilita sinais de temporização deste valor até 500 KHz/65536 programáveis pela CPU. Os endereços para acesso aos temporizadores estão na tabela 4.2.

A PCDD possui um controlador de interrupção que possui oito canais, sendo que três estão acoplados aos temporizadores internos descritos anteriormente, três acoplados a temporizadores externos advindos através do barramento principal, um no conversor A/D para indicar final de conversão e um está de reserva para futuras expansões.

- Bloco 9

É o módulo que seleciona os dispositivos de entrada e saída (E/S) da PCDD. Os endereços decodificados estão dispostos na tabela 4.2.

- Bloco 10

Este bloco é composto por três portas de E/S programáveis como entrada ou saída. Os endereços das portas, bem como o de sua programação estão na tabela 4.2.

- Blocos 11 e 12

A PCDD possui um conversor A/D e um D/A de 10 bits. A entrada do conversor A/D é multiplexada em oito canais que são selecionados através da porta de E/S 08h.

Tabela 4.2 - Endereço dos dispositivos de E/S

Dispositivo de E/S	Endereço
Porta digital A	00h
Porta digital B	01h
Porta digital C	02h
Controle da portas	03h
Porta de saída do D/A (D9 e D8)	04h
Porta de saída do D/A (D7 a D0)	05h
Controle dos canais analógicos	08h
Início de conversão A/D	09h
Leitura de dados A/D (D0 a D7)	0Ah
Leitura de dados A/D (D9 e D8)	0Bh
Canal 0 do temporizador	0Ch
Canal 1 do temporizador	0Dh
Canal 2 do temporizador	0Eh
Controle do temporizador	0Fh
Controle 1 de interrupção	10h
Controle 2 de interrupção	11h

4.6.2 - Utilização do sistema

O CPR1 oferece ao usuário uma série de facilidades de desenvolvimento de programas para controle das juntas.

Foram desenvolvidos no laboratório dois programas (apresentados no Apêndice A) para testes de uma junta do manipulador:

- Programa monitor da PCDD: este programa é residente na EPROM e coordena todo o sistema da PCDD. Está escrito em linguagem Assembler para o Z80. Para melhor entendimento de seu funcionamento, a figura 4.22 mostra seu fluxograma.

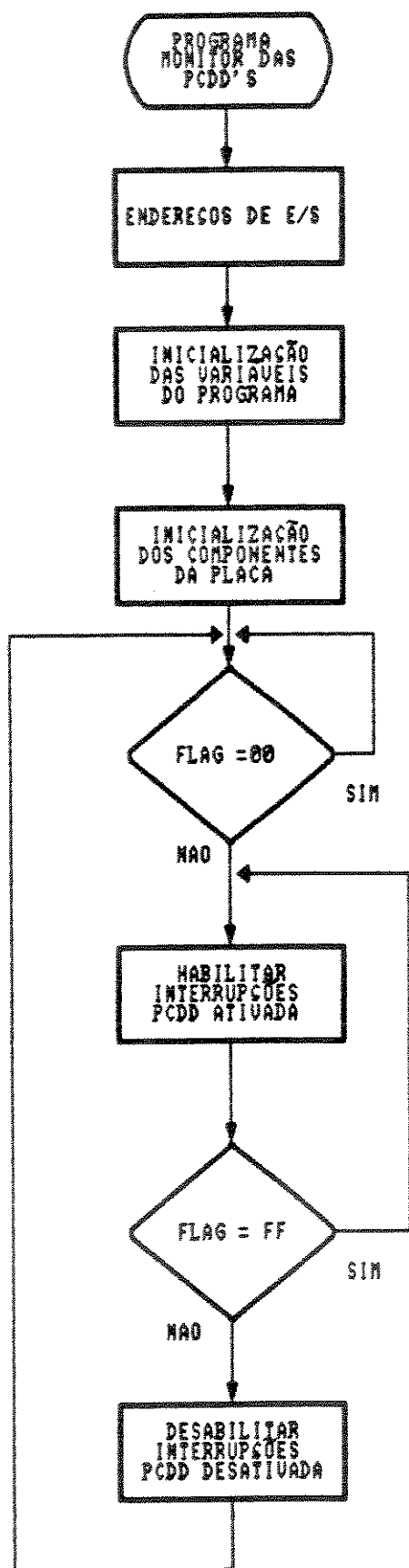
- Programa de interface homem/máquina: este programa faz a interface entre o sistema e o usuário que deseja controlar a junta. Nele estão contidos diversas opções, tais como:

- Envio de um programa completo para a junta;
- Envio dos parâmetros de um regulador PID;
- Envio da posição de referência;
- Habilitação para a junta executar o programa de controle;
- Verificação do comportamento da resposta da junta através de gráficos.

Com estes dois programas, o procedimento básico de desenvolvimento de programas de controle de junta pode ser resumido nas seguintes etapas:

- Desenvolver o programa de controle em um montador (no caso de desenvolvimento em linguagem Assembler) ou um compilador que trabalhe no Micro Mestre mas simule um ambiente do Z80 (comercialmente existem vários programas desse gênero);

- Após obter o programa desenvolvido e colocá-lo em um arquivo tipo .hex, executar o programa de interface homem/máquina;



OBSERVAÇÕES :

1) FLAG = FF INDICA QUE O PROGRAMA MONITOR DEVE PERMITIR A EXECUÇÃO DA ROTINA DE CONTROLE

2) FLAG = 00 INDICA QUE O PROGRAMA MONITOR NÃO DEVE PERMITIR A EXECUÇÃO DA ROTINA DE CONTROLE.

Figura 4.22 - Fluxograma do programa monitor da PCDD

- Enviar o programa para a junta que se deseja controlar;
- Enviar os parâmetros do regulador projetado, posição de referência, e habilitar a execução;
- Se desejar verificar o desempenho da junta, chamar o pacote gráfico do programa.

4.7 - Sensores

Como foi visto no capítulo 3, o controle de rigidez necessita de realimentações das variáveis de posição e de força, desta forma, é necessário a utilização de sensores para a medição dessas duas grandezas. Também é feito a medição de duas outras variáveis para possíveis implementações de algoritmos diferentes, são elas: velocidade do motor e corrente de armadura do motor.

Para medição dessas quatro variáveis foram utilizados os seguintes sensores:

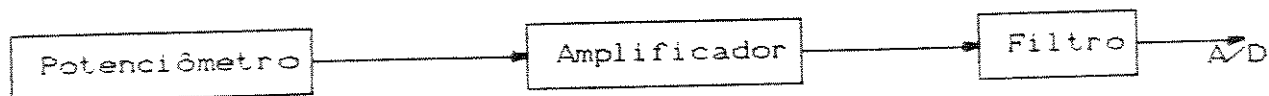
- Potenciômetro: medição de posição da garra;
- Tacômetro: medição da velocidade do motor;
- Resistor "shunt": medição da corrente de armadura do motor;
- "Strain-Gauge": medição da força de preensão da garra.

Os diagramas de bloco dos circuitos de tratamento dos sinais provenientes desses sensores para entrada na parte de conversão analógico/digital são mostrados na figura 4.23.

4.8 - Sumário

Foi visto neste capítulo os aspectos construtivos e os aspectos de modelagem da garra mecânica de dois dedos proposta neste trabalho.

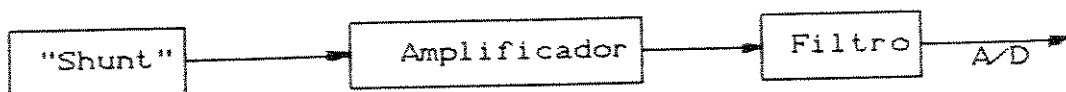
No capítulo a seguir será feito a implementação do controle de rigidez com os base nos resultados deste capítulo.



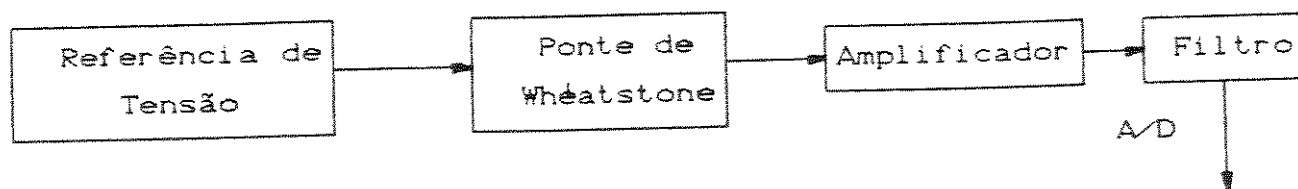
(a)



(b)



(c)



(d)

Figura 4.23 - Diagramas de bloco dos circuitos de tratamento de sinais dos sensores:

- (a) - Potenciômetro
- (b) - Tacômetro
- (c) - Resistor "Shunt"
- (d) - "Strain-Gauge"

CAPÍTULO 5

IMPLEMENTAÇÃO DO CONTROLE DE RIGIDEZ APLICADO NA GARRA DE DEDO FIXO CONSTRUÍDA

5.1 - Introdução

Este capítulo tem como objetivo projetar o controlador de rigidez, apresentado no capítulo 3, aplicado a garra de dedo fixo construída no laboratório, cuja descrição foi apresentada no capítulo 4.

Como o modelo dinâmico da garra encontrado apresenta não linearidades, foi utilizado um método heurístico para a obtenção dos parâmetros do controlador.

Desta forma, o primeiro passo foi realizar uma simulação do modelo encontrado, cujos resultados são apresentados na seção 5.2.

Realizada a simulação e escolhido o ganho ideal para o controlador, que no caso é a constante de rigidez desejada, foi feita a implementação do algoritmo de controle, cuja descrição do "software" de controle é apresentada na seção 5.3.

Finalmente foi feita, na seção 5.4, uma análise do desempenho do controlador de rigidez analisando os resultados obtidos na seção anterior. Nessa análise procurou-se obter conclusões qualitativas sobre a viabilidade e desempenho do algoritmo de controle adotado.

5.2 - Simulação do Modelo Dinâmico da Garra com Controle de Rigidez

Com o modelo do sistema da garra apresentado no capítulo 4, pode-se simular o controle de rigidez apresentado no capítulo 3. A simulação foi feita no programa TUTSIM de projetos de engenharia e otimização por simulação de sistemas dinâmicos contínuos. Este programa foi escolhido por diversos fatores, dentre eles destaca-se a ótima relação facilidade de utilização/desempenho.

Os parâmetros utilizados nessa simulação foram obtidos da seguinte forma:

- Motor cc: os parâmetros do motor foram obtidos pela utilização do método de identificação de parâmetros de motores cc a imã permanente proposto em [Lord 1977]. Os parâmetros foram obtidos de acordo com a teoria apresentada no Apêndice B;

- Transmissão: a rigidez da correia dentada e do cabo de aço foram obtidos experimentalmente através de ensaio em uma máquina de tração, no Instituto de Física da UNICAMP;

- Pinça, objeto e a célula de carga: os valores da rigidez equivalente para alguns materiais podem ser encontrados em [Parker & Paul 1987];

Força de atrito: os valores máximos foram estimados por observação visual do comportamento da garra e foi utilizado uma aproximação de que ela diminua com o quadrado da velocidade.

A listagem do programa do TUTSIM para simular o controle de rigidez pode ser encontrada no apêndice C.

5.3 - Implementação do Algoritmo de Controle de Rigidez

A implementação do controlador de rigidez é feita no sistema de controle digital descrito no capítulo anterior. Os programas da Interface Homem-Máquina e do Monitor da PCDD fornecerão a estrutura básica para a execução do programa de controle.

O programa de controle desenvolvido é colocado na rotina de interrupção da PCDD, na prioridade 1. Esta rotina de interrupção será executada sempre que existir um pedido de interrupção dado pelo canal de temporização. No bloco de temporização será programado o tempo de amostragem do algoritmo de controle.

A figura 5.1 apresenta o fluxograma do algoritmo de controle implementado em Assembler do Z80.

Esse fluxograma apresenta os seguintes blocos:

-Amostragem dos sinais da posição e da força: Nesse bloco é feito a leitura dos sinais de posição e de força através do conversor A/D. O sinal da posição está ligado ao canal 1 do bloco conversor A/D e a força no canal 2;

-Armazenagem dos valores do sinal de posição na tabela 1 e de força na tabela 2: A posição e força são armazenados em tabelas para que se possa verificar posteriormente a evolução desses sinais durante a execução do programa de controle. Essas tabelas são lidas pelo Micro Mestre através do programa de Interface Homem-Máquina e apresentadas em gráficos para análise do usuário;

-Conversão dos sinais amostrados do código offset binário para o código sinal+módulo: O conversor A/D fornece como saída um sinal codificado em offset binário, que apresenta a seguinte forma:

00h	—————→	5V
01h	—————→	$5V - 5V / 128$
02h	—————→	$5V - 2.(5V / 256)$
.		.
.		.
.		.
FEh	—————→	$5V - 255.(5V / 128)$
FFh	—————→	-5V

No entanto, este não é um código normalmente usado para operações aritméticas. O código utilizado foi o sinal+módulo que apresenta um byte para representação do sinal do número, 00h para valores positivos e FFh para valores negativos, e um byte para o módulo do número, da seguinte forma:

00h	—————→	0V
01h	—————→	5V / 128
02h	—————→	5V / 127
.		.
.		.
.		.
80	—————→	5V

-Cálculo do erro de posição;

-Multiplicação do erro de posição pela constante de rigidez;

-Cálculo do erro de força;

Conversão do código sinal+módulo para offset binário: Essa conversão é feita porque o bloco gerador do sinal MLP do recortador trabalha com o código offset binário;

-Envio de dados para o Gerador do sinal MLP.

Esse programa está listado no Apêndice A.

É necessário fazer algumas observações importantes com relação aos problemas vinculados a implementação do algoritmo de controle, são elas:

- Sinal da posição: O sinal que chega a PCDD tem uma excursão de -5V a +5V, o que corresponde a um curso de 48 mm na garra. Como foi descrito no capítulo 4, a PCDD possui conversores de 10 bits, no entanto serão utilizados apenas 8 bits na conversão para que ruídos inerentes às montagens experimentais não ocasionem perturbações no controle.

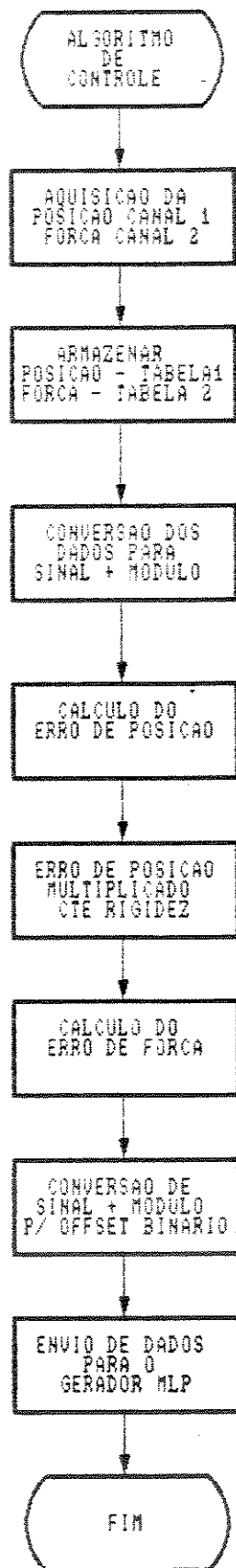


Figura 5.1 : Fluxograma do Algoritmo de Controle Implementado

Assim a precisão que o sistema terá com relação a posição será de:

$$\frac{48 \text{ mm}}{256} = 0,1875 \text{ mm}$$

-Sinal de Força: A excursão do sinal de força é de 0V a 5V. Assim, será utilizado apenas metade da precisão do conversor A/D (com oito bits). Mesmo assim a precisão é razoável, sendo dada por:

$$\frac{189 \text{ N}}{128} = 1,4766 \text{ N}$$

-Erro de posição: o erro de posição, segundo o diagrama de blocos da figura 3.10, é dado por:

$$e = x_d - x \quad (5.1)$$

Para não existir saturação no sinal da força comandada é necessário que esta não ultrapasse seu valor máximo, assim:

$$e \cdot K_\theta \leq f_{\text{máx}} \quad (5.2)$$

-Sistema de representação em ponto fixo: O algoritmo foi implementado com o sistema de numeração em ponto fixo com tamanho de um byte. Isso ocasiona o problema de só poder dar ganhos inteiros no controle do sistema, neste caso a constante de rigidez só terá valores inteiros

Constante de rigidez: O valor real da constante de rigidez é dado por:

$$K_{\text{real}} = K_\theta \cdot 7874,6 \text{ N/m} \quad , \text{ para } K_\theta = 1, 2, \dots, 256.$$

Isso é devido a quantificação dos valores da posição e da força.

5.4 - Resultados Experimentais

Com o programa de controle implementado e com essas observações feitas foram realizados testes na plataforma de teste da garra.

Inicialmente, foi implementado um controlador proporcional de posição com ganho unitário para que fosse obtido medidas de referência.

Primeiro, a garra é comandada para ir da posição 20 mm para 10 mm, com o espaço entre as pinças livre. A resposta pode ser vista na figura 5.2.

Depois, com um objeto encostado na pinça fixa, a garra é comandada para ir da posição 20 mm para 10 mm, com controle proporcional de posição. Esse teste foi feito para dois objetos diferentes: aço e borracha. As figuras 5.3 (a) e (b) mostram a resposta da força e posição, respectivamente, deste teste para o objeto de aço. A figura 5.4 (a) e (b) mostram a resposta da força e posição, respectivamente, para o objeto de borracha.

Feito isso, o programa de controle de rigidez é carregado e pode-se iniciar os teste para o controle de rigidez.

O teste padrão possui as seguintes características:

- Posição inicial = 20 mm;
- Posição de referência = 10 mm;

Foram feitos testes para o objeto de aço e para o de borracha com ganhos $k_e = 1$ e $K_e = 2$. Os resultados foram os seguintes:

- Objeto de aço e $K_e = 1$. Figura 5.5 (a) e (b) para respostas de força e posição, respectivamente;
- Objeto de aço e $K_e = 2$. Figura 5.6 (a) e (b) para respostas de força e posição, respectivamente;
- Objeto de borracha e $K_e = 1$. Figura 5.7 (a) e (b) para respostas de força e posição, respectivamente;

- Objeto de borracha e $K\theta = 2$. Figura 5.8 (a) e (b) para respostas de força e posição, respectivamente.

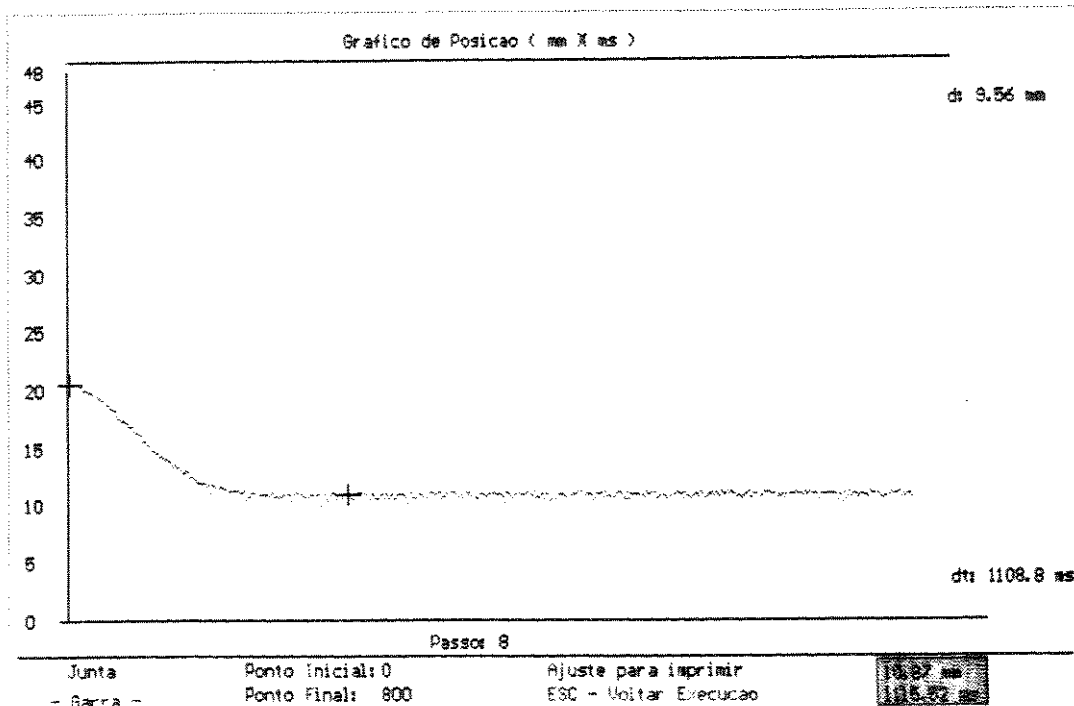
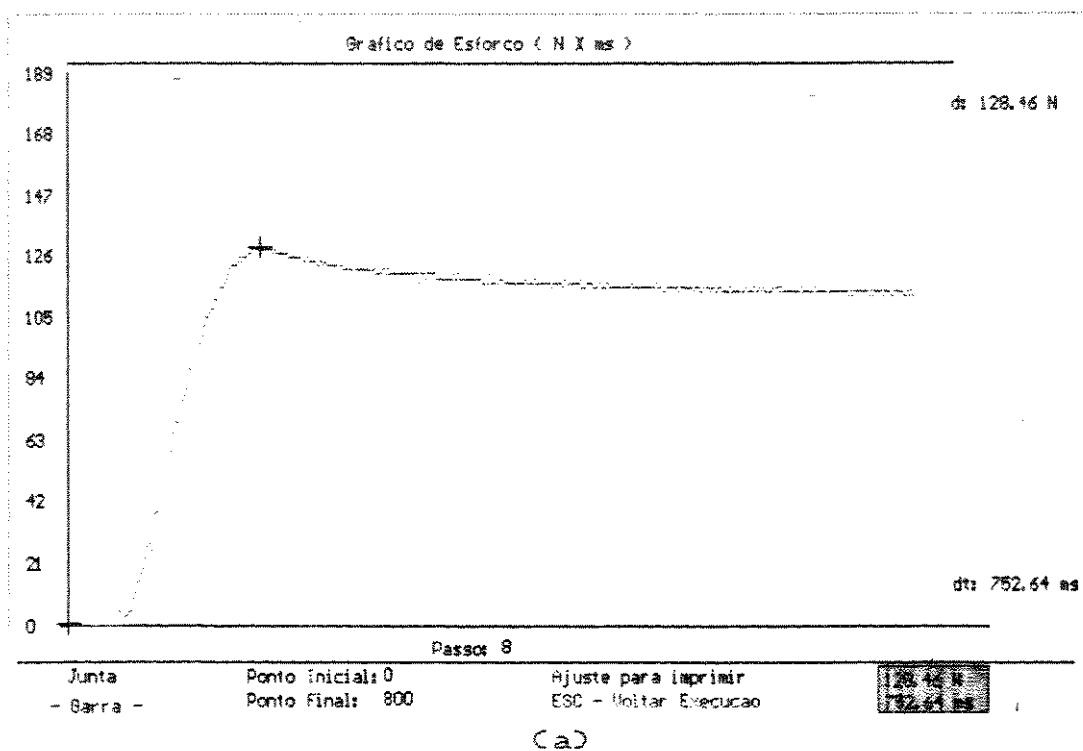
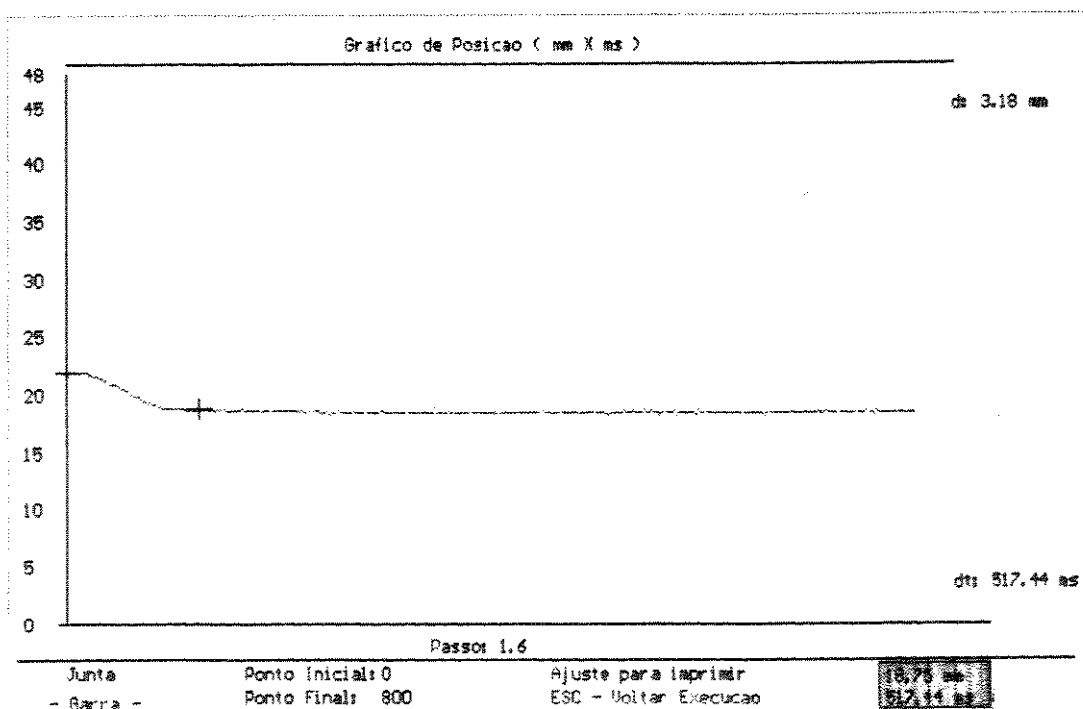


Figura 5.2 - Gráfico da Posição x Tempo para um movimento efetuado da posição 20 mm para 10 mm.

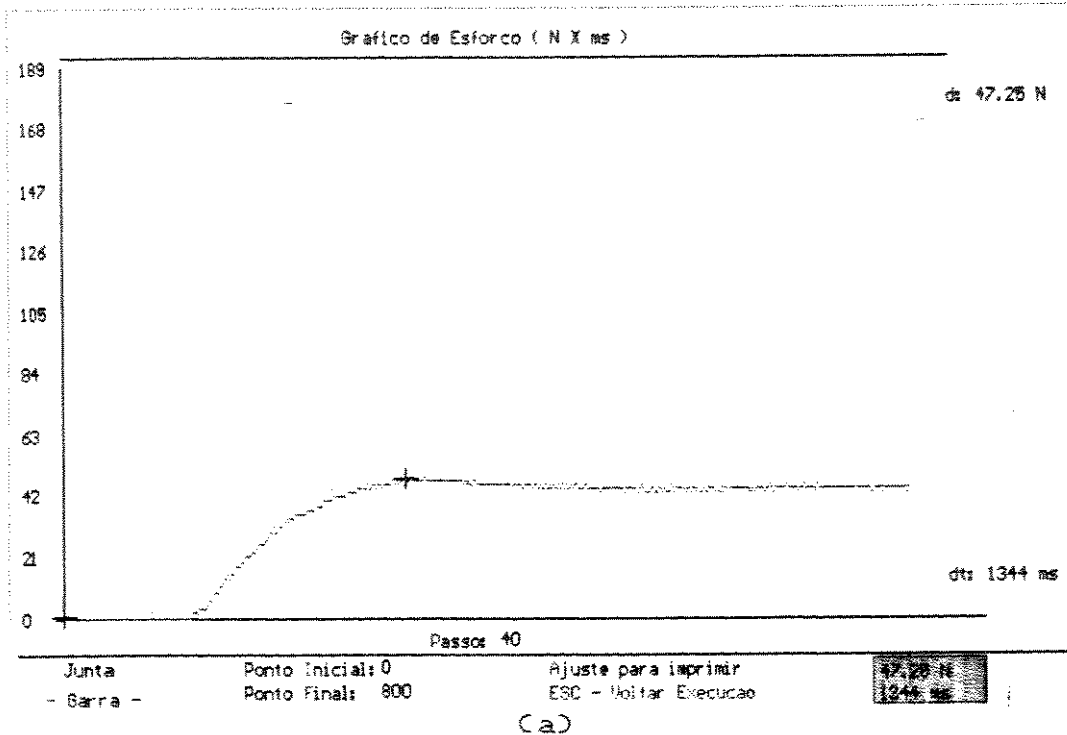


(a)

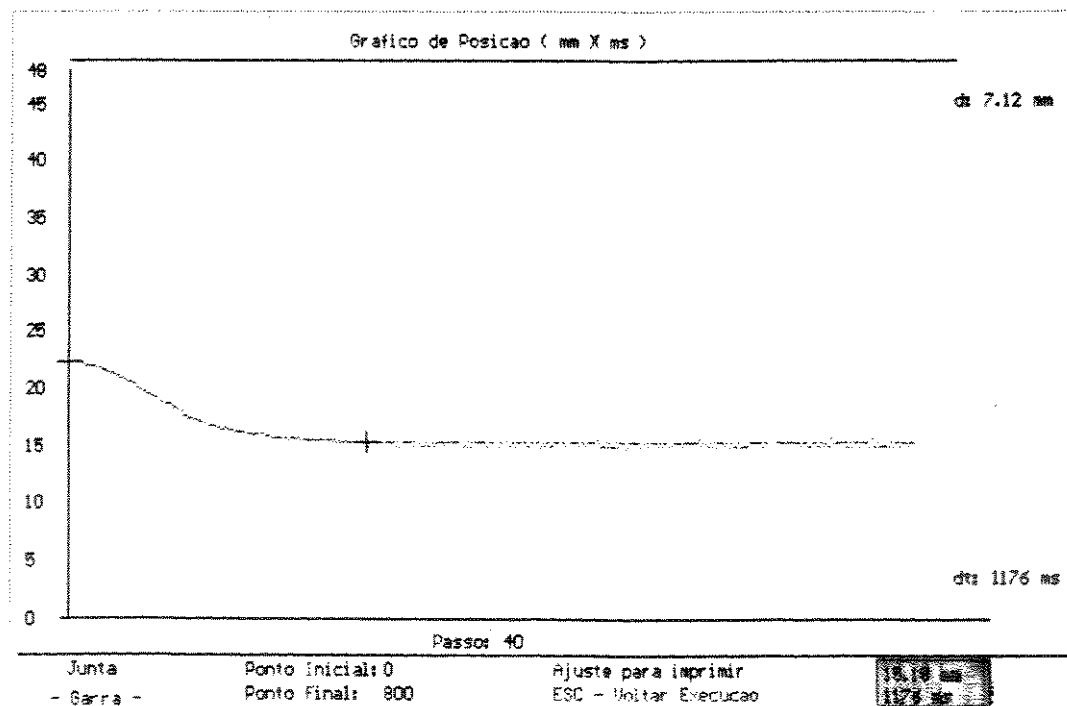


(b)

Figura 5.3 - (a) Resposta da Força com objeto de aço encostado na pinça.
 (b) Resposta de Posição com objeto de aço encostado na pinça.



(a)



(b)

Figura 5.4 : (a) Resposta da Força com objeto de borracha encostado na pinça
(b) Resposta de Posição com objeto de borracha encostado na pinça

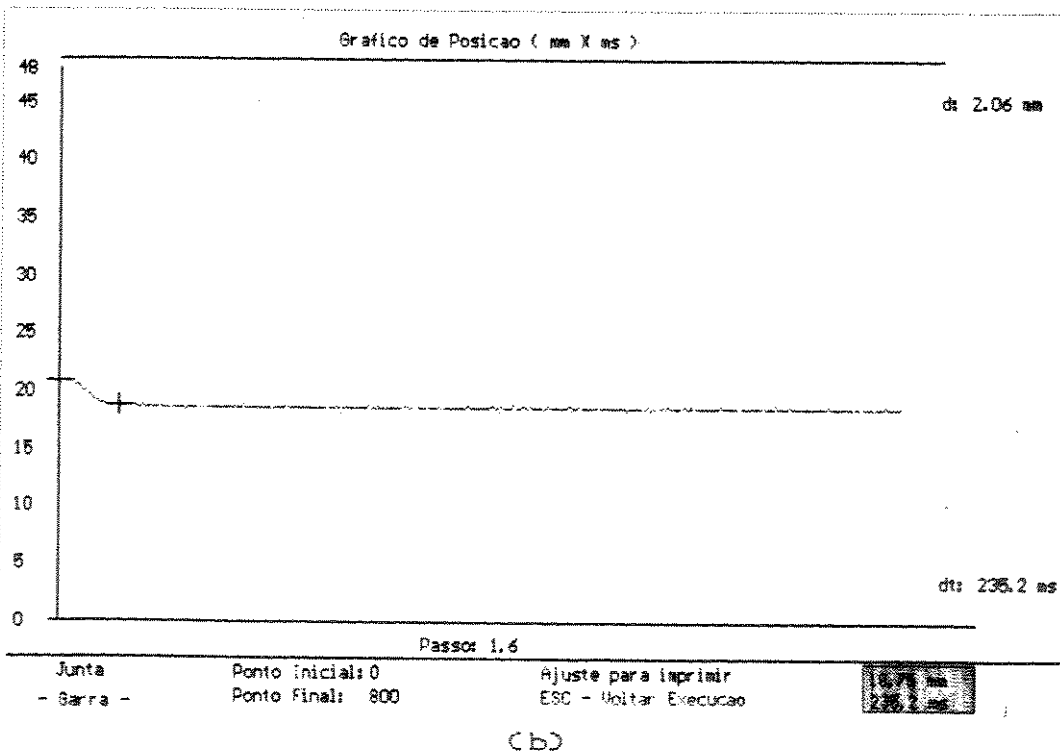
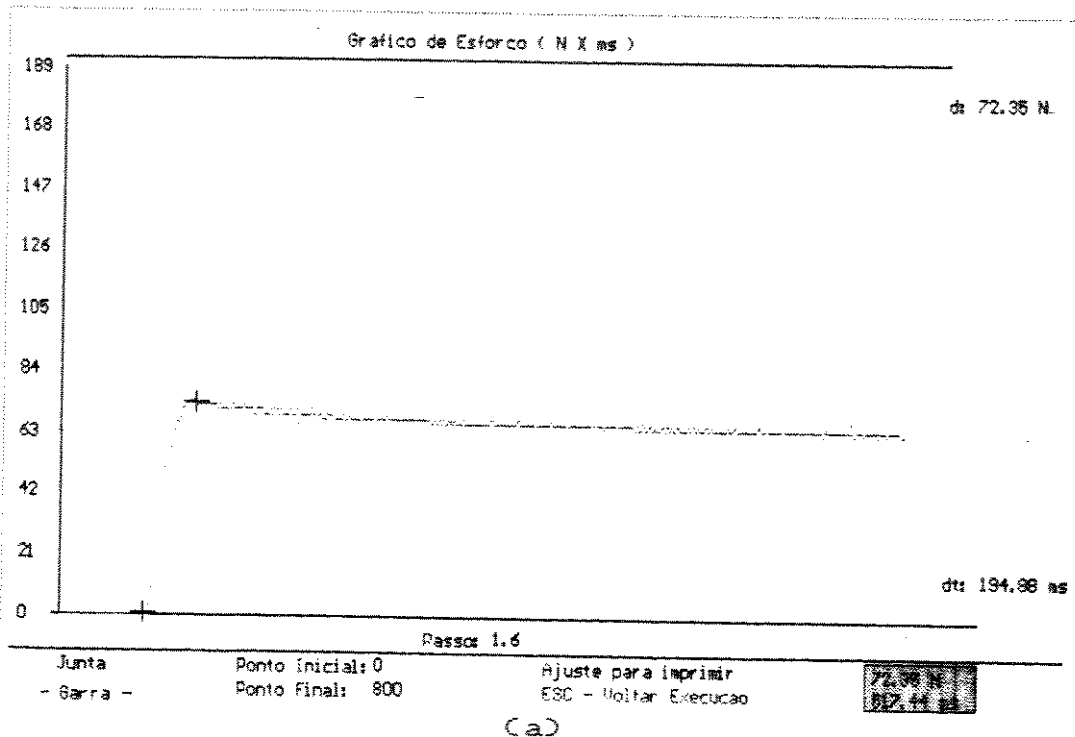
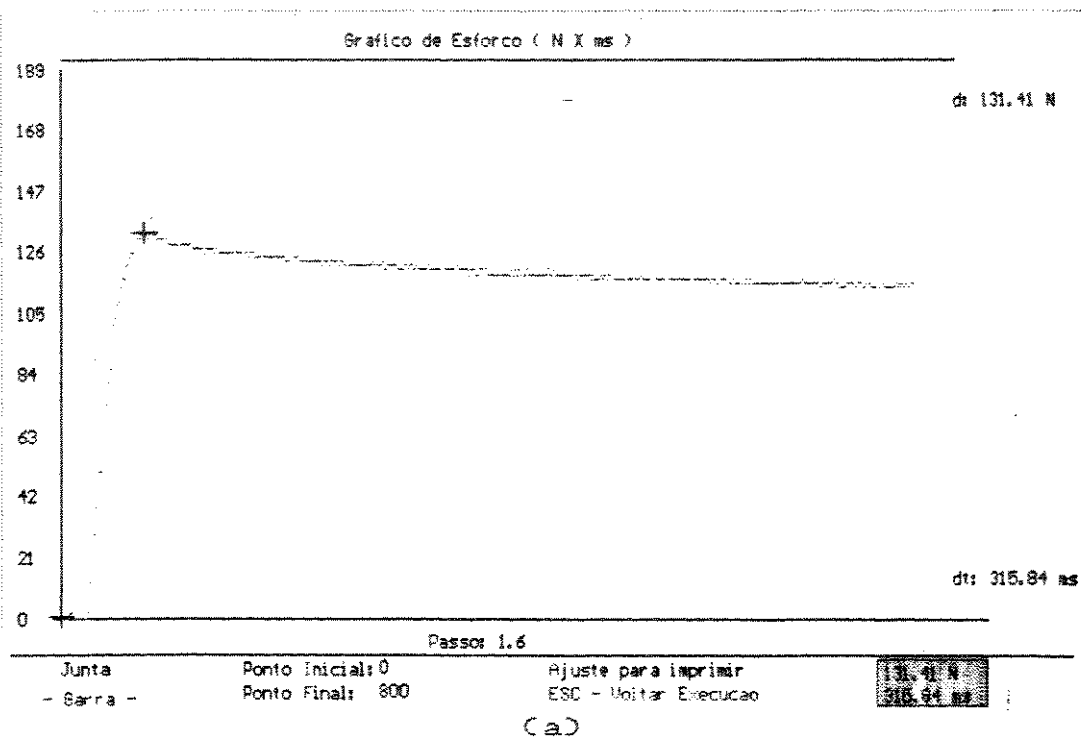
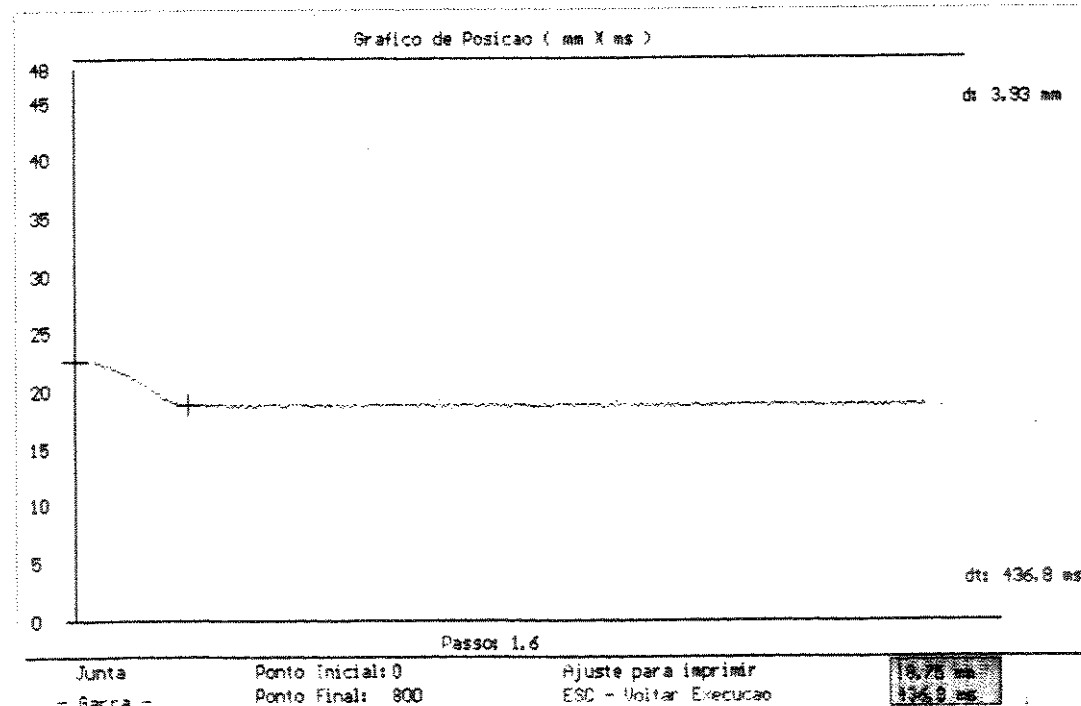


Figura 5.5 - (a) Gráfico de Esforço x tempo para movimento com o Controle de rigidez instalado com ganho $k_e = 1$ e objeto de aço encostado a pinça.

(b) Gráfico de Posição x tempo para movimento com o Controle de rigidez instalado com ganho $k_e = 1$ e objeto de aço encostado a pinça.



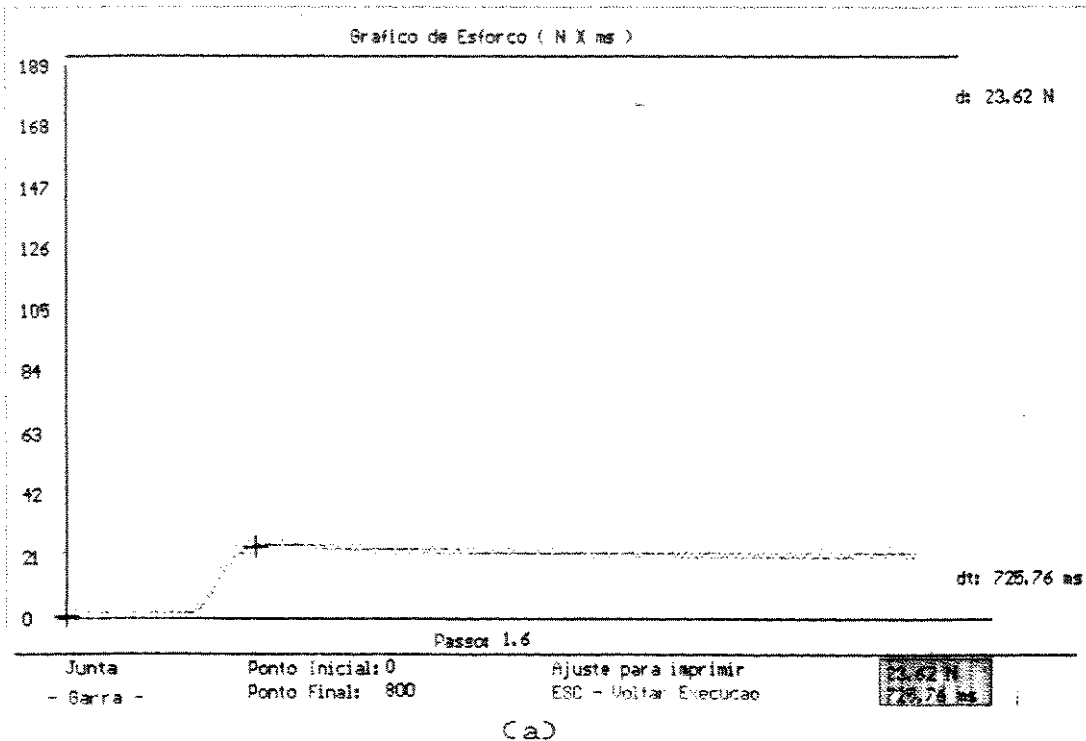
(a)



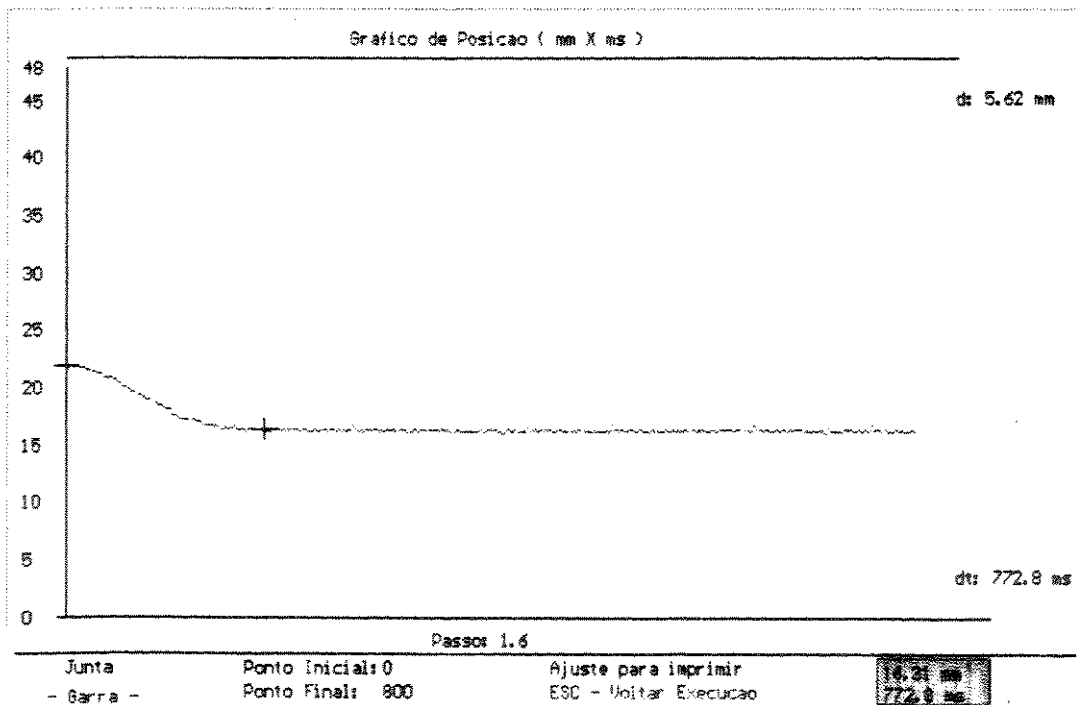
(b)

Figura 5.6 - (a) Gráfico de Esforço x tempo para movimento com o Controle de rigidez instalado com ganho $k_e = 2$ e objeto de aço encostado a pinça.

(b) Gráfico de Posição x tempo para movimento com o Controle de rigidez instalado com ganho $k_e = 2$ e objeto de aço encostado a pinça.



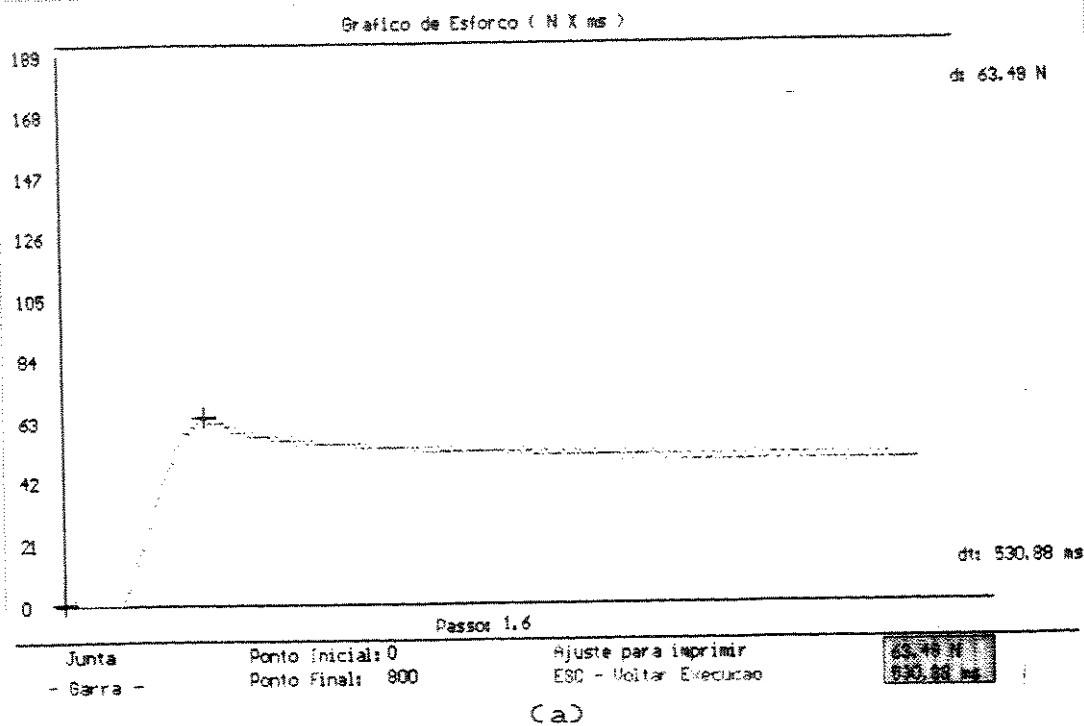
(a)



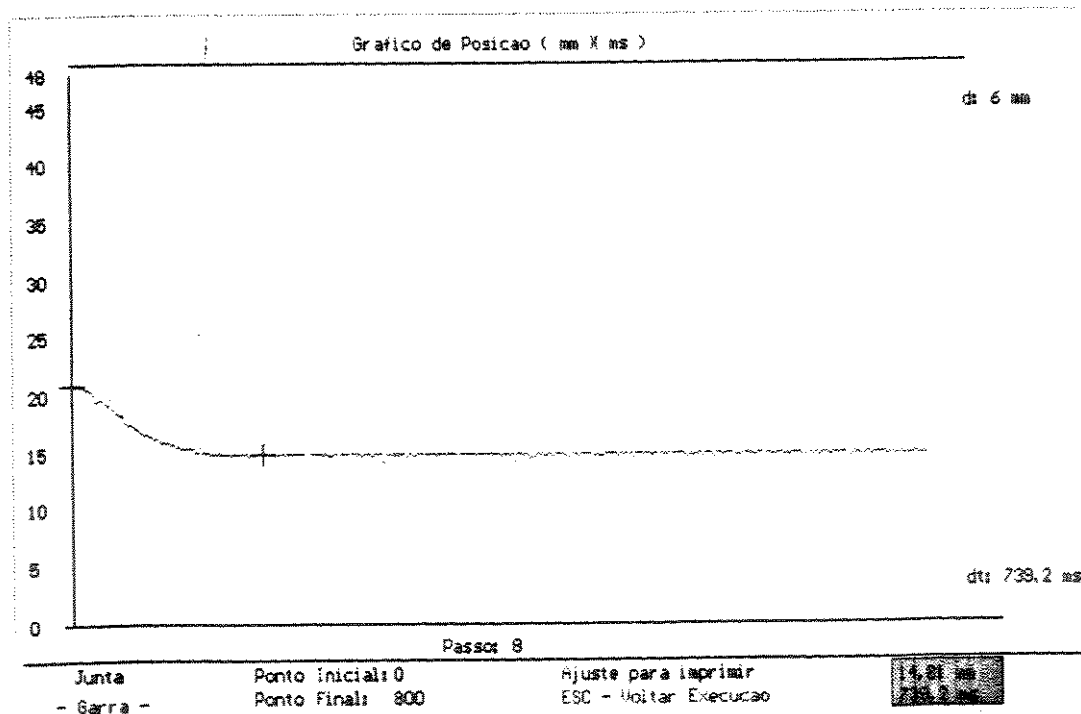
(b)

Figura 5.7 : (a) Gráfico de Esforço x tempo para movimento com o Controle de rigidez instalado com ganho $k_\theta = 1$ e objeto de borracha encostado a pinça.

(b) Gráfico de Posição x tempo para movimento com o Controle de rigidez instalado com ganho $k_\theta = 1$ e objeto de borracha encostado a pinça.



(a)



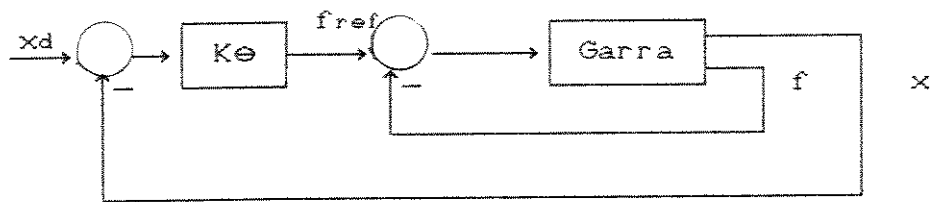
(b)

Figura 5.8 : (a) Gráfico de Esforço x tempo para movimento com o Controle de rigidez instalado com ganho $k_e = 2$ e objeto de borracha encostado a pinça.

(b) Gráfico de Posição x tempo para movimento com o Controle de rigidez instalado com ganho $k_e = 2$ e objeto de borracha encostado a pinça.

5.5 - Análise dos Resultados Obtidos

Para analisar os resultados obtidos deve-se lembrar que o controle de rigidez visa o controle de forças que surgem decorrentes do erro de posição. Este fato fica evidente no diagrama de blocos da figura 3.10, que é redesenhada abaixo.



Assim, pode-se estimar o valor da força medida pelo sensor, pois ela tenderá seguir a força de referência. Esta última pode ser calculada da seguinte forma:

$$f_{ref} = K_{\theta} \cdot (x_d - x) \quad (5.1)$$

A seguir será analisado cada caso comparando a força de referência com a força medida. Os dados necessários são retirados dos gráficos 5.2 a 5.8 e o valor de K_{θ} do capítulo 3:

- Figura 5.5 : Erro de posição = 7,5 mm
Força de referência = 59 N
Força medida (experimental) = 63 N
- Figura 5.6.: Erro de posição = 5,63 mm
Força de referência = 88,7 N
Força medida (experimental) = 110 N
- Figura 5.7.: Erro de posição = 3.94 mm
Força de referência = 31 N
Força medida (experimental) = 22 N

- Figura 5.8.: Erro de posição = 3.58 mm
Força de referência = 56 N
Força medida (experimentalmente) = 48 N

Como pode ser observado, os valores da força medida diferem da força de referência. Isso deve-se ao fato de não ter sido considerado, na figura 3.10, as forças de atrito consideradas no capítulo 4.

Comparando os gráficos das figuras 5.2 a 5.8, pode-se observar que o controle de rigidez realmente atua no valor da força sobre o objeto.

5.6 - Sumário

Foi visto neste capítulo aspectos relacionados à implementação do controle de rigidez, tais como: simulação do modelo dinâmico da garra, algoritmo de controle, problemas relacionados a implementação do algoritmo de controle e testes.

Os resultados foram satisfatórios, mostrando a viabilidade da utilização do controle de rigidez em garras mecânicas.

CAPÍTULO 6

CONCLUSÕES

Neste trabalho foi apresentado um estudo criterioso sobre problemas relacionados com garras mecânicas articuladas.

Realizou-se um estudo sobre a operação de preensão por mãos humanas visando obter conhecimentos mais profundos sobre os aspectos relacionados com a operação de preensão de objetos.

Um estudo da análise cinemática e de forças possibilitou a análise de diversos mecanismos que podem restringir completamente um objeto.

Também foi realizado um estudo sobre o estado da arte em controle de força e posição utilizados em robótica para a execução de tarefas onde existe um contato entre o manipulador e o meio. Deste estudo, foi estudado com mais detalhes o controle de rigidez, por se adequar melhor ao controle de garras mecânicas.

Feito todos esses estudos, foi projetada e implementada uma garra mecânica de dois dedos, sendo um fixo. Essa garra está inserida no projeto JECA II, que é um manipulador que possui seis graus de liberdade.

Com a garra construída, foi implementado o algoritmo de controle de rigidez.

Com todas as etapas concluídas pode-se fazer algumas considerações:

- MECANISMO

O mecanismo construído apresentou-se bastante robusto e com um grau de precisão bastante razoável. O sistema montado apresenta grandes não linearidades, principalmente na transmissão através do fuso, o que pode ser minorado através do uso de fuso com esferas recirculantes.

Deve ser melhorado o acoplamento entre os mecanismos de transmissão, principalmente entre o cabo de aço e a pinça móvel.

- CONTROLE

O controle de rigidez apresentou bons resultados que encorajam a continuidade de sua utilização. Os principais problemas que existiram foram:

- Implementação do algoritmo em ponto fixo - Neste tipo de implementação o valor da constante de rigidez recebe valores muito distanciados (7874,6 - 15749,2 - etc), e isto leva a um controlador com muito pouco ajuste. Isso deve ser resolvido com a implementação do algoritmo em ponto flutuante, no entanto deve-se ter o cuidado com o tempo de execução do algoritmo, que irá crescer muito.

- Não linearidade do mecanismo - As não linearidades do mecanismo são grandes, assim o controle de rigidez deve ser adaptado para sistemas não lineares. Para minorar esse problema, pode ser colocado um sensor de proximidade para que não exista um impacto muito grande entre a garra e o objeto, pois é esse impacto que causa grandes não linearidades no sistema.

SENSORES

Os sensores apresentaram um desempenho razoável, pois como o sistema trabalha com sistemas chaveados para acionamento dos motores, existe muito ruído no sistema, o que leva a uma conversão com menos precisão que o conversor A/D pode fornecer. Uma solução para esse problema é a utilização de codificadores óticos para a medição de

posição e velocidade. Pelo mesmo motivo, a medição de corrente também deve ser melhorada, e um sensor mais adequado para essas situações é o sensor de efeito Hall.

Com relação ao sensor de força, deve ser melhorado o substrato na qual foram afixados os "strain-gauges", para que não influencie na dinâmica do sistema.

Com essas considerações, pode-se concluir que os objetivos iniciais foram plenamente e satisfatoriamente alcançados.

BIBLIOGRAFIA

- ALONSO e FINN. Curso de Física - Vol. 1. Editora Guanabara Dois. Rio de Janeiro 1978.
- ÁLVAREZ, Adelheid I.M..Caos em Sistemas de Controle de Motores CC pela Técnica PLL/Dual. Dissertação de Mestrado - FEE - UNICAMP. Campinas, 1991.
- ARAÚJO, Aluísio F.R..Caos em Sistemas de Controle com Modelos Discretos. Dissertação de Mestrado - FEE - UNICAMP. Campinas, 1988.
- CABRERA, M. A..Análise do Controle de Impedância Aplicado ao Controle de Esforços em Tarefas de Contato. Dissertação de Mestrado - FEE - UNICAMP. Campinas, 1990.
- CUTKOSKY, M.R..On Grasp Choice, grasp Models and the Design of Hands for Manufacturing Tasks. IEEE Tansactions on Robotics and Automation Vol.5 no.3. Junho de 1989.
- DIAS, Marcus de A..Controlador Programável a Multimicroprocessadores para Controle de Hierárquico de Robôs. Dissertação de Mestrado - FEE - UNICAMP. Campinas, 1991.
- Eletro-Craft Corp..DC Motor Speed Controls Servo Systems. Minnesota, 1973.
- FERRATÉ, G..Robótica Industrial. Marcombo Editores. Barcelona, 1986.
- HANAFUSA, H. e ASADA, H..Stable Prehension by a Robot Hand with Elastic Fingers. 8th International Symposium on Industrial Robots. Tokyo, 1977.

- HANAFUSA, H e ASADA, H..A Robot Hand with Elastic Fingers and its Application to Assembly Processes. IFAC Symposium on Information and Control Problems in Manufacturing Technology. Tokyo, 1978.
- HOGAN, N..Impedance Control: An Approach to Manipulation - Part I - Theory. ASME Journal of Dynamic Systems, Measurement, and Control vol.107. Março de 1985.
- LIPKIN, H e DUFFY, J..Analysis of Industrial Robots Via the Theory of Screws. Proceedings of 12th International Symposium on Industrial Robots. Paris, 1982.
- LORD. Parameters Identification of DC Motors. IEEE Transactions on Industrial Electronics. Junho, 1977.
- MADRID, Marconi K.. Controle Autônomo de Trajetórias para Robôs Manipuladores. Dissertação de Doutorado (em andamento) - FEE - UNICAMP. Campinas, 1991.
- MASON, M..Compliance and Force Control for Computer Controlled Manipulators. IEEE Transactions on Systems, Man, and Cybernetics vol.SMC-11. Junho de 1981.
- NAPIER, J..A mão do Homem: Anatomia, Função, Evolução. Editora Universidade de Brasília / Zahar Editores. Rio de Janeiro 1983.
- OHWOVORIOLE, E.N..Kinematics and Friction in Grasping by Robotic Hands. ASME Journal of Mechanisms, Transmissions and Automation in Design vol.109. Setembro de 1987.
- OHWOVORIOLE, M.S. e ROTH, B..An Extension of Screw Theory. ASME Journal of Mechanical Design vol.103. Outubro de 1981.
- PALHARES, Álvaro G.B..Notas de Aula do Curso de Pós-Graduação da FEE/UNICAMP do curso de Controle por Chaveamento. Agosto de 1988.

- PARKER, J.K. e PAUL, F.W..Controlling Impact Forces in Pneumatic Robot Hand Designs. ASME Journal Dynamic Systems, Measurement, and Control vol 109. Dezembro de 1987.
- RAIBERT, M.H. e CRAIG, J.J..Hybrid Position/Force Control of Manipulators. ASME Journal of Dynamic Systems, Measurement and Control vol.102. Junho de 1981.
- SALISBURY, J.K..Active Stiffness Control of a Manipulator in Cartesian Coordinates. Proceedings of IEEE Decision and Control Conference. Albuquerque, 1980.
- SALISBURY, J.K..Kinematic and Force Analysis of Articulated Hands. Dissertação de Doutorado. Stanford 1982.
- SALISBURY, J.K..Kinematic and Force Analysis of Mechanical Hands. ASME Journal of Mechanisms, Transmission and Automation Design vol. 105. Março de 1983.
- VOSSOUGH, R. e DONATH, M..Robot Finger Stiffness Control in the Presence of Mechanical Nonlinearities. ASME Journal of Dynamic Systems, Measurement and Control vol. 110. Setembro de 1988.
- WHITNEY, D.E..Force Feedback Control of Manipulator Fine Motions. Journal of Dynamic Systems, Measurement and Control. Junho de 1977.
- WHITNEY, D.E..Historical Perspective and State of the Art in Robot Force Control. IEEE Proceedings of International Conference on Robotics and Automation 1985.

Adilson Sakahi Ohfugi

APÊNDICES

da tese de mestrado

Garras Articuladas para Robôs Manipuladores : Análise cinemática e de
Forças, Sensoreamento e Controle de posição e Esforços na
Preensão de Objetos

Orientador: Prof. Dr. Álvaro Geraldo Badan Palhares

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA
DEPARTAMENTO DE SISTEMAS E CONTROLE DE ENERGIA

Campinas - SP
Agosto 1991

A P Ê N D I C E A

"SOFTWARE UTILIZADO"

```

/*                                     Campinas,19/07/91      */
/*                                     */
/*                                     */
/*     Este Programa Faz Parte do Pacote para Controlar Trajetorias */
/* de Robos-Manipuladores Mecanicos.                                */
/*     Desenvolvido por :  Marconi Kohl Madrid                     */
/*                                     Adilson Sakahi Ohfugi         */
/*                                     */
/*     Finalidade: Testar controle PID da garra atraves do PC      */
/*                                     */

#include <dos.h>
#include <dir.h>
#include <math.h>
#include <conio.h>
#include <stdio.h>          /* Inclui as bibliotecas necessarias */
#include <stdlib.h>
#include <stdarg.h>
#include <graphics.h>
#include <process.h>
#include <string.h>
#include "gc.h"

char *Fonts[] = {
    "DefaultFont",    "TriplexFont",    "SmallFont",
    "SansSerifFont", "GothicFont"
};

char *LineStyle[] = {
    "SolidLn",    "DottedLn",    "CenterLn",    "DashedLn",    "UserBitLn"
};

char *FillStyles[] = {
    "EmptyFill",    "SolidFill",    "LineFill",    "LtSlashFill",
    "SlashFill",    "BkSlashFill",    "LtBkSlashFill", "HatchFill",
    "XHatchFill",    "InterleaveFill", "WideDotFill",    "CloseDotFill"
};

char *TextDirect[] = {
    "HorizDir",    "VertDir"
};

char *HorizJust[] = {
    "LeftText",    "CenterText",    "RightText"
};

char *VertJust[] = {
    "BottomText",    "CenterText",    "TopText"
};

```

```

int      GraphDriver; -      /* Tipo de driver grafico      */
int      GraphMode;         /* Valor do modo grafico      */
double   Raspecto;          /* Razao de aspecto dos pixels na tela */
int      MaxX, MaxY;         /* Maxima Resolucao da tela  */
int      MaxColors;          /* Maximo de cores avaliaveis */
int      ErrorCode;          /* Codigo de erros graficos    */
struct palettetype palette; /* Usado para ler informacoes do palette */

int      Grav[800];
char     prog[60];
int      *aux, *aux1, *errsim;
int      indpgr;

struct PTS {
    int x, y;
}; /* Estrutura para segurar pontos de vertice */

/*
/*      Comeco da Funcao Principal
/*
int main()
{
    int i;
    Inicializa(); /* Seta sistema no modo grafico */
    if( GraphDriver==EGA || GraphDriver==EGALO || GraphDriver==VGA )
        indpgr= 0;
    Tela();
    for(i = 1; i == 8; i++)
        Grav[i]= 0x00;
    IniciaPia();
    Menu();

    closegraph(); /* Retorna o sistema ao mode texto */
    return(0);
}

/*
/*      INICIALIZA: Inicializa o sistema grafico e mostra qualquer
/*      erro ocorrido.
/*
void Inicializa(void)
{
    int xasp, yasp; /* Usado para leitura da razao de aspecto */

    GraphDriver = DETECT; /* Requisita auto-detecao */
    initgraph( &GraphDriver, &GraphMode, "C:/USR/BIN/TC" );
    ErrorCode = graphresult(); /* Le resultado da inicializacao */
    if( ErrorCode != grOk ){ /* Erro ocorrido durante inicio */
        printf( " Erro no sistema Grafico: %s\n", grapherrormsg(ErrorCode));
        exit( 1 );
    }
}

```

```

getpalette( &palette ); - /* Le o palette de trabalho */
MaxColors = getmaxcolor() + 1; /* Le o maximo numero de cores */
MaxX = getmaxx();
MaxY = getmaxy(); /* Le o tamanho da tela */

getaspectratio(&xasp,&yasp ); /* Le a razao de aspecto do hardware */
Raspecto = (double)xasp/(double)yasp; /* Da o fator de correcao */
}

/*****
/*      VIOLETA: Injecao do programa para execucao na junta      */
/*      */
*****/

int Violeta(void)
{
    int Cace(char *str);
    int Caca(char *str);

    FILE *fp;
    char ch1,ch2[2],ch4[4];
    int count,end,chk,chk1,i,l;
    unsigned char res,mem[36];

    int PortA, PortC, Pcont;
    int Cont1, SporA, SporC;
    int check;

    PortA = 0x220;
    PortC = 0x222;
    Pcont = 0x223;

    Cont1 = 0x82;
    SporA = 0xFF;
    SporC = 0x0F;

    outportb(Pcont,Cont1);
    outportb(PortC,SporC);
    outportb(PortA,SporA);
    if((fp = fopen(prog,"rb")) == NULL){
        setcolor(LIGHTRED);
        outtextxy(MaxX/4+30,MaxY*3/4+5,"Programa de Junta Inexistente");
        sound(3000);
        delay(200);
        nosound();
        delay(1500);
        setcolor(BLACK);
        outtextxy(MaxX/4+30,MaxY*3/4+5,"Programa de Junta Inexistente");
        setfillstyle(SOLID_FILL,BLUE);
        bar(MaxX/4+35,MaxY*3/4-12,MaxX/4+300,MaxY*3/4-2);
        bar(MaxX/4+25,MaxY/4+3,MaxX*3/4-25,MaxY/4+30);
        return(1);
    }
}

```

```

check= 0;
while (2 == 2){
    ch1 = getc(fpp);
    if (ch1 != ':'){
        setcolor(LIGHTRED);
        outtextxy(MaxX/4+30,MaxY*3/4+5,"Programa de junta em
        formato errado");
        sound(3000);
        delay(200);
        nosound();
        delay(1500);
        setcolor(BLACK);
        outtextxy(MaxX/4+30,MaxY*3/4+5,"Programa de junta em
        formato errado");
        setfillstyle(SOLID_FILL,BLUE);
        bar(MaxX/4+35,MaxY*3/4-12,MaxX/4+300,MaxY*3/4-2);
        bar(MaxX/4+25,MaxY/4+3,MaxX*3/4-25,MaxY/4+30);
        return(1);
    }
    ch2[0]= getc(fpp);
    ch2[1]= getc(fpp);
    count= Caca(ch2);
    chk= count;
    ch4[0]= getc(fpp);
    ch4[1]= getc(fpp);
    ch4[2]= getc(fpp);
    ch4[3]= getc(fpp);
    end= Caca(ch4);
    chk += Caca(ch4) + Caca(ch4+2);
    ch2[0]= getc(fpp);
    ch2[1]= getc(fpp);
    chk += Caca(ch2);
    if(Caca(ch2) == 1)
        return (0);
    for (i=0 ; i<count; i++) {
        ch2[0]= getc(fpp);
        ch2[1]= getc(fpp);
        mem[i]= Caca(ch2);
        chk += mem[i];
    }
    ch2[0]= getc(fpp);
    ch2[1]= getc(fpp);
    chk1= (0xFF-(chk&0x00FF)+1);
    if(Caca(ch2) == chk1){
        check= check;
    }
    else{
        check= 1;
    }
    l=0;
    for(i=0 ; i<count; i++) {
        while (l != 10){
            outportb(0x220,0x00);
            outportb(0x222,0x46);
            pokeb(0x8000,end,mem[i]);
            outportb(0x222,0x0F);
            outportb(0x220,0xFF);

```

```

        outportb(0x220, 0x00);
        outportb(0x222, 0x46);
        res = peekb(0x8000, end);
        outportb(0x222, 0x0F);
        outportb(0x220, 0xFF);
        if (res != mem[i]) {
            l = l + 1;
        } else {
            end++;
            break;
        }
    }
    if (l >= 10) {
        setcolor(LIGHTRED);
        outtextxy(MaxX/4+30, MaxY*3/4+5, "Erro de gravacao na memoria");
        sound(3000);
        delay(500);
        nosound();
        setcolor(BLACK);
        outtextxy(MaxX/4+30, MaxY*3/4+5, "Erro de gravacao na memoria");
        setfillstyle(SOLID_FILL, BLUE);
        bar(MaxX/4+35, MaxY*3/4-12, MaxX/4+300, MaxY*3/4-2);
        bar(MaxX/4+25, MaxY/4+3, MaxX*3/4-25, MaxY/4+30);
        return(1);
    }
    ch2[0] =getc(fp);
    ch2[1] =getc(fp);

    setfillstyle(SOLID_FILL, BLUE);
    bar(MaxX/4+35, MaxY*3/4-12, MaxX/4+300, MaxY*3/4-2);
    bar(MaxX/4+25, MaxY/4+3, MaxX*3/4-25, MaxY/4+30);
    settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
    bar(90, MaxY-15, 400, MaxY-5);
    setcolor(LIGHTRED);
    outtextxy(100, MaxY-12, prog);
    setcolor(LIGHTGREEN);
    outtextxy(MaxX/4+80, MaxY*3/4-10, "Junta foi programada");
    setfillstyle(SOLID_FILL, BLUE);
    bar(MaxX/4+25, MaxY/4+3, MaxX*3/4-25, MaxY/4+30);
    settextstyle(SMALL_FONT, HORIZ_DIR, 4);
    indpgr = 1;
}
fclose(fp);
return(0);
}

```

```

/*****
/*      CACE: Funcao de Violeta
/*
/*****
int Cace(str)
char *str;
{
    return(256*Caca(str) + Caca(str+2));
}

/*****
/*      CACA: Funcao de Violeta
/*
/*****
int Caca(str)
char *str;
{
    int Decod(char str);

    return (16*Decod(str[0]) + Decod(str[1]));
}

/*****
/*      DECOD: Funcao de Caca
/*
/*****
int Decod(chr)
char chr;
{
    int a ;
    switch (chr) {
    case '0':
        a=0;
        break;
    case '1':
        a=1;
        break;
    case '2':
        a=2;
        break;
    case '3':
        a=3;
        break;
    case '4':
        a=4;
        break;
    case '5':
        a=5;
        break;
    case '6':
        a=6;
        break;

```

```

case '7':
    a=7;
    break;
case '8':
    a=8;
    break;
case '9':
    a=9;
    break;
case 'A':
    a=10;
    break;
case 'B':
    a=11;
    break;
case 'C':
    a=12;
    break;
case 'D':
    a=13;
    break;
case 'E':
    a=14;
    break;
case 'F':
    a=15;
    break;    }
return (a);
}

/*****
/*      TELA: Coloca Tela Principal      */
/*                                          */
/*****/

void Tela(void)
{
    setfillstyle(SOLID_FILL, BLACK);
    bar(0,0,MaxX,MaxY);
    setcolor(WHITE);
    rectangle(0,0,MaxX,MaxY);
    setfillstyle(SOLID_FILL, BLUE);
    bar(MaxX/4, MaxY/4, MaxX*3/4, MaxY*3/4);
    rectangle(MaxX/4, MaxY/4, MaxX*3/4, MaxY*3/4);
    setfillstyle(SOLID_FILL, LIGHTBLUE);
    bar(MaxX/4+50, MaxY*3/4-35, MaxX*3/4-50, MaxY*3/4-15);
    setcolor(LIGHTGREEN);
    settextstyle(DEFAULT_FONT, HORIZ_DIR, 2);
    settextjustify(CENTER_TEXT, TOP_TEXT);
    outtextxy(MaxX/2, 20, "Jeca II");
    settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
    outtextxy(MaxX/2, 60, "Controle de Posicao e Esforco da Garra");
}

```

```

outtextxy(MaxX/2,75,"Controlador PID");
settextjustify(LEFT_TEXT,TOP_TEXT);
setcolor(LIGHTCYAN);
outtextxy(MaxX/4+30,MaxY/4+40,"Controlador");
setcolor(GREEN);
line(MaxX/4+25,MaxY/4+50,MaxX/4+125,MaxY/4+50);
setcolor(LIGHTGREEN);
outtextxy(MaxX/4+40,MaxY/4+60,"      P");
outtextxy(MaxX/4+40,MaxY/4+80,"      I");
outtextxy(MaxX/4+40,MaxY/4+100,"     D");
outtextxy(MaxX/4+65,MaxY*3/4-28,"      A");
setcolor(LIGHTCYAN);
outtextxy(MaxX/4+40,MaxY/4+60,"Const  :");
outtextxy(MaxX/4+40,MaxY/4+80,"Const  :");
outtextxy(MaxX/4+40,MaxY/4+100,"Const  :");
Amostra();
setcolor(BLACK);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
outtextxy(MaxX/4+65,MaxY*3/4-28,"Posicao  tual:");
setcolor(LIGHTBLUE);
outtextxy(30,MaxY-55,"F10");
outtextxy(400,MaxY-55,"F1");
outtextxy(400,MaxY-40,"    F12");
outtextxy(400,MaxY-25,"    ESC");
outtextxy(30,MaxY-40,"F9");
outtextxy(30,MaxY-25,"F4");
setcolor(WHITE);
settextjustify(LEFT_TEXT,TOP_TEXT);
outtextxy(30,MaxY-55,"    - Programar Junta");
outtextxy(400,MaxY-55,"    - Executar");
outtextxy(400,MaxY-40,"    - Como Utilizar");
outtextxy(400,MaxY-25,"    - Sair");
outtextxy(30,MaxY-40,"    - Verificar Comunicacao");
outtextxy(30,MaxY-25,"    - Simular");
if(indpgr == 0){
setcolor(LIGHTRED);
outtextxy(MaxX/4+40,MaxY*3/4-10,"** Junta nao foi programada **");
}else{
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
bar(90,MaxY-15,400,MaxY-5);
setcolor(LIGHTRED);
outtextxy(100,MaxY-12,prog);
setcolor(LIGHTGREEN);
outtextxy(MaxX/4+80,MaxY*3/4-10,"Junta foi programada");
setfillstyle(SOLID_FILL,BLUE);
bar(MaxX/4+25,MaxY/4+3,MaxX*3/4-25,MaxY/4+30);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
}
}

```

```

/*****
/*      MENU:
/*
/*****

void Menu(void)
{
#define precisao      0.1875

    int patual;
    int l,le,i,c;
    int ximp,yimp,errsim,aux;
    int sinal,modulo;
    float result,gresult,rp;
    float intei,frac;

    int SM;
    int res;
    char num[10];
    char sinter[10];

    patual = 0;
    *aux1 = 0;
    SM= 0x7600;
    rp= 0.0;

    while (2 == 2){
        c= getch();
        if(c == ESC)
            break;

        if(c == F10){
            setfillstyle(SOLID_FILL,LIGHTGRAY);
            bar(MaxX/4+25,MaxY/4+3,MaxX*3/4-25,MaxY/4+30);
            settextstyle(SMALL_FONT,HORIZ_DIR,4);
            setcolor(BLUE);
            outtextxy(MaxX/4+115,MaxY/4+5,"Nome do Programa");
            line(MaxX/4+28,MaxY/4+10,MaxX/4+105,MaxY/4+10);
            line(MaxX/4+28,MaxY/4+10,MaxX/4+28,MaxY/4+28);
            line(MaxX/4+28,MaxY/4+28,MaxX*3/4-28,MaxY/4+28);
            line(MaxX/4+220,MaxY/4+10,MaxX*3/4-28,MaxY/4+10);
            line(MaxX*3/4-28,MaxY/4+10,MaxX*3/4-28,MaxY/4+28);
            setcolor(LIGHTBLUE);
            outtextxy(MaxX/4+40,MaxY/4+16,"*. HEX");
            Identi();
            if(*aux1 == 1){
                setfillstyle(SOLID_FILL,BLUE);
                bar(MaxX/4+25,MaxY/4+3,MaxX*3/4-25,MaxY/4+30);
            }else{
                Violeta();
            }
        }
    }
}

```

```

if(c == LETP)
    patual = 2;
if(c == LETI)
    patual = 3;
if(c == LETD)
    patual = 4;
if(c == F1){
    Telexec();
    while(2 == 2){
        c = getch();
        if(c == ESC){
            Tela();
            break;
        }
        if(c == F5){
            le = 0;
            while(le != 6){ /* Desabilita Rotina de Controle */
                outportb(0x220, 0x00);
                outportb(0x222, 0x46);
                pokeb(0x8000, 0x7506, 0x00);
                outportb(0x222, 0x0F);
                outportb(0x220, 0xFF);

                outportb(0x220, 0x00);
                outportb(0x222, 0x46);
                res = peekb(0x8000, 0x7506);
                res = res & 0x00FF;
                outportb(0x222, 0x0F);
                outportb(0x220, 0xFF);
                if (res != 0x00){
                    le = le + 1;
                }
                else {
                    le = 0;
                    break;
                }
            }
            if(le >= 5){
                setttextjustify(CENTER_TEXT, TOP_TEXT);
                setcolor(LIGHTRED);
                outtextxy(MaxX/2-20, MaxY*3/4+10, "Erro de Comunicacao");
                setttextjustify(LEFT_TEXT, TOP_TEXT);
                delay(100);
                nosound();
                delay(800);
                setfillstyle(SOLID_FILL, BLACK);
                bar(MaxX/2-130, MaxY*3/4+5, MaxX/2+125, MaxY*3/4+20);
            }
            else{
                setcolor(RED);
                circle(450, 110, 5);
                circle(450, 110, 4);
                circle(450, 110, 3);
            }
        }
    }
}

```

```

if(c == LETP){
    setcolor(WHITE);
    rectangle(MaxX/4+255,MaxY/4+37,MaxX/4+305,MaxY/4+52);
    setfillstyle(SOLID_FILL,RED);
    ximp= MaxX/4+257;
    yimp= MaxY/4+40;
    bar(ximp-1,yimp-2,ximp+45,yimp+8);
    Entrad(ximp,yimp,rp,&errsim,&aux,&result);
    if(errsim == 1){
        sound(3000);
        setttextjustify(LEFT_TEXT,TOP_TEXT);
        setcolor(LIGHTRED);
        outtextxy(MaxX/2-20,MaxY*3/4+10,"Fora do Limite");
        delay(800);
        setcolor(BLACK);
        outtextxy(MaxX/2-20,MaxY*3/4+10,"Fora do Limite");
        nosound();
        setfillstyle(SOLID_FILL,RED);
        bar(ximp-1,yimp-2,ximp+45,yimp+8);
        setcolor(RED);
        rectangle(MaxX/4+255,MaxY/4+37,MaxX/4+305,MaxY/4+52);
        setcolor(WHITE);
    }
    else{
        if(result >= 24.0){ /* 24.0 = garra no centro */
            Grav[1]= 0x00;
            gresult= (result-24.0) / precisao;
        }else{
            gresult= (24.0-result)/precisao;
            Grav[1]= 0xFF;
        }
        intei= (int)(gresult);
        frac= gresult - intei;
        if(10*frac > 5.0)
            intei= intei + 1;
        gcvt(intei,4,sinter);
        Grav[2]= atoi(sinter);
        setcolor(RED);
        rectangle(MaxX/4+255,MaxY/4+37,MaxX/4+305,MaxY/4+52);
        bar(ximp-1,yimp-2,ximp+45,yimp+8);
        setcolor(RED);
        circle(450,110,5);
        circle(450,110,4);
        circle(450,110,3);
        setcolor(LIGHTGREEN);
        circle(450,110,6);
        setcolor(WHITE);
        sprintf(num,"%g",result);
        strcat(num," mm");
        outtextxy(ximp,yimp,num);
        rp= result;
    }
}

```

```

if(c == LETA){
    l=0;
    for (i=1 ; i<=8; i++) {
        while (l!=10){ /*Envia Referencia e Ganhos */
            outportb(0x220,0x00);
            outportb(0x222,0x46);
            pokeb(0x8000,SM+i-1,Grav[i]);
            outportb(0x222,0x0F);
            outportb(0x220,0xFF);

            outportb(0x220,0x00);
            outportb(0x222,0x46);
            res=peekb(0x8000,SM+i-1);
            res= res & 0x00FF;
            outportb(0x222,0x0F);
            outportb(0x220,0xFF);
            if (res!=Grav[i]){
                l=l+1;
            }
            else {
                l= 0;
                break;
            }
        }
    }
    l=0;
    while (l!=10){ /* Habilita Escrita na Memoria */
        outportb(0x220,0x00); /* Da Junta para grafico */
        outportb(0x222,0x46);
        pokeb(0x8000,0x7619,0xFF);
        outportb(0x222,0x0F);
        outportb(0x220,0xFF);
        outportb(0x220,0x00);
        outportb(0x222,0x46);
        res=peekb(0x8000,0x7619);
        res= res & 0x00FF;
        outportb(0x222,0x0F);
        outportb(0x220,0xFF);
        if (res!= 0xFF){
            l=l+1;
        }
        else {
            l= 0;
            break;
        }
    }
    l=0;
    while (l!=10){ /* Atualiza Ponteiro Indicador*/
        outportb(0x220,0x00); /*Inicio tabela para grafico*/
        outportb(0x222,0x46);
        pokeb(0x8000,0x761A,0x00); /* Menos significativo */
        outportb(0x222,0x0F);
        outportb(0x220,0xFF);
    }
}

```

```

    outportb(0x220,0x00);
    outportb(0x222,0x46);
    res=peekb(0x8000,0x761A);
    res= res & 0x00FF;
    outportb(0x222,0x0F);
    outportb(0x220,0xFF);
    if (res!= 0x00){
        l=l+1;
    }
    else {
        l= 0;
        break;
    }
}
l=0;
while (l!=10){
    outportb(0x220,0x00); /* Atualiza Ponteiro Indicador */
    outportb(0x222,0x46); /*Inicio tabela para grafico */
    pokeb(0x8000,0x761B,0x50); /* Mais Significativo */
    outportb(0x222,0x0F);
    outportb(0x220,0xFF);
    outportb(0x220,0x00);
    outportb(0x222,0x46);
    res=peekb(0x8000,0x761B);
    res= res & 0x00FF;
    outportb(0x222,0x0F);
    outportb(0x220,0xFF);
    if (res!= 0x50){
        l=l+1;
    }
    else {
        l= 0;
        break;
    }
}
le= 0;
while(le!=6){
    outportb(0x220,0x00);
    outportb(0x222,0x46);
    pokeb(0x8000,0x7506,0xFF);
    outportb(0x222,0x0F);
    outportb(0x220,0xFF);
    outportb(0x220,0x00);
    outportb(0x222,0x46);
    res=peekb(0x8000,0x7506);
    res= res & 0x00FF;
    outportb(0x222,0x0F);
    outportb(0x220,0xFF);
    if (res!= 0xFF){
        le=le+1;
    }
}

```

```

        else {
            le= 0;
            break; }
    }
    if((cl >= 10)!(cle >=5)){
        setttextjustify(CENTER_TEXT,TOP_TEXT);
        setcolor(LIGHTRED);
        outtextxy(MaxX/2-20,MaxY*3/4+10,"Erro de Comunicacao");
        setttextjustify(LEFT_TEXT,TOP_TEXT);
        delay(100);
        nosound();
        delay(800);
        setfillstyle(SOLID_FILL,BLACK);
        bar(MaxX/2-130,MaxY*3/4+5,MaxX/2+125,MaxY*3/4+20);
    }
    else{
        setcolor(LIGHTGREEN);
        circle(450,110,6);
        circle(450,110,5);
        circle(450,110,4);
        circle(450,110,3);
    }
}
}
if(c == NUM1){
    Grafm();
}
if(c == NUM2){
    Grafforc();
}
}
}
setcolor(LIGHTRED);
if(patual == 2){
    rectangle(MaxX/4+130,MaxY/4+55,MaxX/4+180,MaxY/4+70);
    ximp= MaxX/4+140;
    yimp= MaxY/4+58;
    Entra(ximp,yimp,&errsim,&aux,&sinal,&modulo);
    Grav[3]= sinal;
    Grav[4]= modulo;
    setfillstyle(SOLID_FILL,BLUE);
    bar(ximp+1,yimp-2,ximp+35,yimp+8);
    if(Errsim == 0){
        if(aux == 0)
            Amostra(); }
    setcolor(BLUE);
    rectangle(MaxX/4+130,MaxY/4+55,MaxX/4+180,MaxY/4+70);
}
if(patual == 3){
    rectangle(MaxX/4+130,MaxY/4+75,MaxX/4+180,MaxY/4+90);
    ximp= MaxX/4+140;
    yimp= MaxY/4+78;
    Entra(ximp,yimp,&errsim,&aux,&sinal,&modulo);
    Grav[5]= sinal;
    Grav[6]= modulo;
}

```

```

        setfillstyle(SOLID_FILL,BLUE);
        bar(ximp+1,yimp-2,ximp+35,yimp+8);
        if(errsim == 0){
            if(aux == 0)
                Amostra();
        }
        setcolor(BLUE);
        rectangle(MaxX/4+130,MaxY/4+75,MaxX/4+180,MaxY/4+90);
    }
    if(patual == 4){
        rectangle(MaxX/4+130,MaxY/4+95,MaxX/4+180,MaxY/4+110);
        ximp= MaxX/4+140;
        yimp= MaxY/4+98;
        Entra(ximp,yimp,&errsim,&aux,&sinal,&modulo);
        Grav[7]= sinal;
        Grav[8]= modulo;
        setfillstyle(SOLID_FILL,BLUE);
        bar(ximp+1,yimp-2,ximp+35,yimp+8);
        if(errsim == 0){
            if(aux == 0)
                Amostra();
        }
        setcolor(BLUE);
        rectangle(MaxX/4+130,MaxY/4+95,MaxX/4+180,MaxY/4+110);
    }
    patual= 0;
}
}

/*****
/*      ENTRA: Entrada de dados do Teclado      */
/*      */
*****/

void Entra(int ximp,int yimp,int *errsim,int *aux,
           int *sinal,int *modulo)
{
    int c,i;
    char num[10],imp[10];
    int dec[10];

    settextstyle(SMALL_FONT,HORIZ_DIR,4);
    setcolor(WHITE);
    *aux= 0;
    i=1;
    dec[1]=0x00,dec[2]=0x00,dec[3]=0x00;

    sound(1000);
    delay(100);
    nosound();

```

```

while(2 == 2){
    c= getch();
    sound(3000);
    if(c == ESC){
        *errsim= 0;
        *aux= 1;
        nosound();
        break;
    }
    if(c == ENTER){
        delay(100);
        nosound();
        break;
    }
    if(c == DEL){
        setfillstyle(SOLID_FILL,BLUE);
        bar(ximp+1,yimp-2,ximp+35,yimp+8);
        i=1;
        dec[1]=0x00,dec[2]=0x00,dec[3]=0x00;
        delay(100);
        nosound();
        c=getch();
        sound(3000);
    }
    if(i <= 3){
        if((c >= 48) & (c <=57)){
            c= c-48;
            dec[i]= c;
            sprintf(num,"%x",c);
            outtextxy(ximp+i*6,yimp,num);
            i=i+1;
        }
        if(c == LETA){
            dec[i]= 0x0a;
            outtextxy(ximp+i*6,yimp,"a");
            i=i+1;
        }
        if(c == LETB){
            dec[i]= 0x0b;
            outtextxy(ximp+i*6,yimp,"b");
            i=i+1;
        }
        if(c == LETC){
            dec[i]= 0x0c;
            outtextxy(ximp+i*6,yimp,"c");
            i=i+1;
        }
        if(c == LETD){
            dec[i]= 0x0d;
            outtextxy(ximp+i*6,yimp,"d");
            i=i+1;
        }
    }
}

```

```

        if(c == LETE){
            dec[i]= 0x0e;
            outtextxy(ximp+i*6,yimp,"e");
            i=i+1;
        }
        if(c == LETF){
            dec[i]= 0x0f;
            outtextxy(ximp+i*6,yimp,"f");
            i=i+1;
        }
        if(c == 43){
            dec[i]= 0x11;
            outtextxy(ximp+i*6,yimp,"+");
            i=i+1;
        }
        if(c == 45){
            dec[i]= 0x12;
            outtextxy(ximp+i*6,yimp,"-");
            i=i+1;
        }
    }
    delay(100);
    nosound();
}
else{
    sound(5000);
    dec[1]=0x00,dec[2]=0x00,dec[3]=0x00;
    settextjustify(CENTER_TEXT,TOP_TEXT);
    setcolor(LIGHTRED);
    outtextxy(MaxX/2-20,MaxY*3/4+10,"Entrada Ilegal");
    settextjustify(LEFT_TEXT,TOP_TEXT);
    delay(100);
    nosound();
    delay(800);
    setfillstyle(SOLID_FILL,BLUE);
    bar(ximp+1,yimp-2,ximp+35,yimp+8);
    setfillstyle(SOLID_FILL,BLACK);
    bar(MaxX/2-100,MaxY*3/4+5,MaxX/2+25,MaxY*3/4+20);
    i=1;
    setcolor(WHITE);
}
if((i != 4) & (dec[1] != 0x11)){
    if(dec[1] != 0x12){
        sound(5000);
        dec[1]=0x00,dec[2]=0x00,dec[3]=0x00;
        settextjustify(CENTER_TEXT,TOP_TEXT);
        setcolor(LIGHTRED);
        outtextxy(MaxX/2-20,MaxY*3/4+10,"Entrada Ilegal");
        settextjustify(LEFT_TEXT,TOP_TEXT);
        delay(100);
        nosound();
        delay(800);
        setfillstyle(SOLID_FILL,BLUE);
    }
}

```

```

        bar(ximp+1,yimp-2,ximp+35,yimp+8);
        setfillstyle(SOLID_FILL,BLACK);
        bar(MaxX/2-100,MaxY*3/4+5,MaxX/2+25,MaxY*3/4+20);
        i=1;
        setcolor(WHITE);
    }
}

```

```

}
*modulo=0x00;
*errsim= 0;
if((dec[2]==0x11) ! (dec[3]==0x11)!(dec[2]==0x12)!(dec[3]==0x12))<
    *errsim = 1;
}
if(dec[1] == 0x12)<
    *sinal= 0xFF;
    *modulo= dec[3]+ (dec[2] * 0x10);
}
else<
    *sinal= 0x00;
    if(dec[1] == 0x11)<
        *modulo= dec[3] + (dec[2] * 0x10);
    }
    else<
        *modulo= dec[2] + (dec[1] * 0x10);
    }
}
}
}

```

```

/*
/*
/*
/* Este Programa Faz Parte do Pacote para Controlar Trajetorias*/
/* de Robos-Manipuladores Mecanicos. */
/* Rotina Garra 2 */
/*
/*
/* Finalidade: Testar controle PID da garra atraves do PC */

```

```

#include <dos.h>
#include <dir.h>
#include <math.h>
#include <conio.h>
#include <stdio.h>          /* Inclui as bibliotecas necessarias */
#include <stdlib.h>
#include <stdarg.h>
#include <graphics.h>
#include <process.h>
#include <string.h>
#include "gc.h"

extern int      GraphDriver;          /* Tipo de driver grafico*/
extern int      GraphMode;            /* Valor do modo grafico */
extern double   Raspecto;             /* Razao de aspecto dos pixels na tela*/
extern int      MaxX, MaxY;           /* Maxima Resolucao da tela*/
extern int      MaxColors;            /* Maximo de cores avaliaveis*/
extern int      ErrorCode;            /* Codigo de erros graficos*/
extern struct palettetype palette;    /* Usado para ler informacoes
do palette*/

extern int      Grav[800];
extern char     prog[60];
extern int      *aux,*aux1,*errsim;
extern int      indpgr;

struct PTS {
    int x, y;
}; /* Estrutura para segurar pontos de vertice */

```

```

/*****
/*      ENTRAD: Entrada de dados do Teclado
/*
/*
*****/

void EntradCint ximp,int yimp,float rp,int *errsim,int *aux,float
*result)
{
    int c,i,j,jt,ind,deterr;
    char num[10];
    int dec[10];
    float jt1;
    settxtstyle(SMALL_FONT,HORIZ_DIR,4);
    setcolor(WHITE);
    *aux= 0;
    i=1;
    dec[1]=0,dec[2]=0,dec[3]=0,dec[4]=0;
    dec[5]=0,dec[6]=0,dec[7]=0;
    sound(5000);
    delay(100);
    nosound();
    while(2 == 2){
        c= getch();
        sound(5000);
        if(c == ESC){
            *errsim= 0;
            *aux= 1;
            nosound();
            break;
        }
        if(c == ENTER){
            delay(100);
            nosound();
            break;
        }
        if(c == DEL){
            setfillstyle(SOLID_FILL,RED);
            bar(ximp-1,yimp-2,ximp+45,yimp+8);
            i=1;
            dec[1]=0,dec[2]=0,dec[3]=0,dec[4]=0;
            dec[5]=0,dec[6]=0,dec[7]=0;
            delay(100);
            nosound();
            c=getch();
            sound(5000);
        }
        if(i <= 7){
            if((c >= 48) & (c <=57)){
                c= c-48;
                dec[i]= c;
                sprintf(num,"%d",c);
                outtextxy(ximp+i*8,yimp,num);
                i=i+1;
            }
        }
    }
}

```

```

        if(c == 43){
            dec[i] = 11;
            outtextxy(ximp+i*6,yimp,"+");
            i=i+1;
        }

        if(c == 45){
            dec[i] = 12;
            outtextxy(ximp+i*6,yimp,"-");
            i=i+1;
        }

        if(c == 46){
            dec[i] = 10;
            outtextxy(ximp+i*6,yimp,".");
            i=i+1;
        }
    }
    delay(1000);
    nosound();
}
if(i > 6){
    sound(3000);
    dec[1]=0,dec[2]=0,dec[3]=0,dec[4]=0;
    dec[5]=0;

    settextjustify(LEFT_TEXT,TOP_TEXT);
    setcolor(LIGHTRED);
    outtextxy(MaxX/2-20,MaxY*3/4+10,"Entrada Ilegal");
    delay(800);
    setcolor(BLACK);
    outtextxy(MaxX/2-20,MaxY*3/4+10,"Entrada Ilegal");
    nosound();
    setfillstyle(SOLID_FILL,RED);
    bar(ximp-1,yimp-2,ximp+45,yimp+8);
    i=1;
    setcolor(WHITE);
}
}

*errsim= 0;

if((dec[2]==11) | (dec[3]==11) | (dec[4]==11) | (dec[5]==11) |
    (dec[6]==11) | (dec[2]==12) | (dec[3]==12) |
    (dec[4]==12) | (dec[5]==12) | (dec[6]==12)){

    *errsim = 1;
}
if((dec[1]!=10) & (dec[2]!=10) & (dec[3]!=10) &
    (dec[4]!=10)){
    *errsim = 1;
}
}

```

```

deterr=0;
for( j=1 ; j<8 ; ++j ){
    if(dec[j] == 10)
        deterr= deterr+1;
}
if (deterr > 1){
    *errsim= 1;
}

if((dec[1]==12) ! (dec[1]==11)){

    if((dec[6]==10) ! (dec[7]==10)){
        *errsim= 1;
    }
    ind=0;
    for( j=2 ; j<8 ; ++j){
        if(dec[j] == 10)      /* localizacao do pontinho */
            ind= j;
    }
    *result=0.0;
    jt=1;
    if(ind > 2){
        for ( j =ind-1 ; j >= 2 ; --j ){
            *result= *result+ (jt * dec[j]);
            jt= jt*10;
        }
    }
    jt1=0.1;
    for ( j =ind+1 ; j <= ind+2 ; j++ ){
        *result= *result + (jt1 * dec[j]);
        jt1= jt1/10;
    }
    if(dec[1] == 12)
        *result= -1.0 * *result;
}
else
{
    if((dec[5]==10) ! (dec[6]==10) ! (dec[7]==10)){
        *errsim = 1;
    }

    ind=0;
    for( j=1 ; j<8 ; ++j){
        if(dec[j] == 10)      /* localizacao do pontinho */
            ind= j;
    }
    *result=0.0;
    jt=1;
    if(ind > 1){
        for ( j =ind-1 ; j >= 1 ; --j ){
            *result= *result+ (jt * dec[j]);
            jt= jt*10;
        }
    }
}

```

```

        >
        jt1=0.1;
        for ( j =ind+1 ; j <= ind+2 ; ++j )<
            *result = *result + (jt1 * dec[j]);
            jt1= jt1/10;
        >
    >
    if(*aux == 1)
        *errsim = 0;
    if(*result > 48.0)
        *errsim = 1;
    if(dec[ind + 2] != 0)
        *errsim = 1;
    if(*aux == 1)
        *result= rp;
}
/*****
/*      ENTRAT: Entrada de dados do Teclado (Tempo)      */
/*
/*
/*****

void Entrat(int ximp,int yimp,int *t)
{
    int c,i,j,jt1,ind;
    char num[10];
    int dec[10];
    settxtstyle(SMALL_FONT,HORIZ_DIR,4);
    setcolor(WHITE);
    *aux= 0;
    i=1;
    ind= 0;
    dec[1]=0,dec[2]=0,dec[3]=0,dec[4]=0,dec[5]=0;
    sound(1000);
    delay(100);
    nosound();
    while(2 == 2){
        c= getch();
        sound(3000);
        if(c == ESC){
            *errsim= 0;
            *aux= 1;
            nosound();
            break;
        }
        if(c == ENTER){
            nosound();
            break;
        }
    }
}

```

```

if(c == DEL){
    setfillstyle(SOLID_FILL, BLUE);
    bar(ximp+1, yimp-2, ximp+35, yimp+8);
    i=1;
    ind= 0;
    dec[1]=0, dec[2]=0, dec[3]=0, dec[4]=0, dec[5]=0;
    delay(100);
    nosound();
    c=getch();
    sound(3000);
}
if(i <= 5){
    if((c >= 48) & (c <=57)){
        c= c-48;
        dec[i]= c;
        sprintf(num, "%d", c);
        outtextxy(ximp+i*6, yimp, num);
        i=i+1;
        ind= ind + 1;
    }
    delay(100);
    nosound();
}else{
    sound(5000);
    dec[1]=0, dec[2]=0, dec[3]=0, dec[4]=0, dec[5]=0;
    settextjustify(LEFT_TEXT, TOP_TEXT);
    delay(100);
    nosound();
    delay(800);
    i=1;
    ind= 0;
    setcolor(WHITE);
}
}
}
}
}
}

void IniciaPia(void)
{
    int PortA, PortC, Pcont;
    int Cont1, SporA, SporC;
    PortA = 0x220;
    PortC = 0x222;
    Pcont = 0x223;
    Cont1 = 0x82;
    SporA = 0xAA;
    SporC = 0x0F;
}

```

```

    outportb(Pcont, Cont1);
    outportb(PortC, SporC);
    outportb(PortA, SporA);
}
/*****
/*      AMOSTRA: Atualiza ganhos na tela
*****/

void Amostra(void)
{
    int p;
    char num[10];
    setfillstyle(SOLID_FILL, BLUE);
    bar(MaxX/4+100, MaxY/4+55, MaxX/4+149, MaxY/4+110);
    setcolor(WHITE);
    settextstyle(SMALL_FONT, HORIZ_DIR, 4);
    if(Grav[3] == 0xFF){
        outtextxy(MaxX/4+105, MaxY/4+58, "-");
        p= Grav[4];
    }
    else{
        outtextxy(MaxX/4+105, MaxY/4+58, "+");
        p= Grav[4];
    }
    sprintf(num, "%x", p);
    outtextxy(MaxX/4+113, MaxY/4+58, num);
    if(Grav[5] == 0xFF){
        outtextxy(MaxX/4+105, MaxY/4+78, "-");
        p= Grav[6];
    }
    else{
        outtextxy(MaxX/4+105, MaxY/4+78, "+");
        p= Grav[6];
    }
    sprintf(num, "%x", p);
    outtextxy(MaxX/4+113, MaxY/4+78, num);
    if(Grav[7] == 0xFF){
        outtextxy(MaxX/4+105, MaxY/4+98, "-");
        p= Grav[8];
    }
    else{
        outtextxy(MaxX/4+105, MaxY/4+98, "+");
        p= Grav[8];
    }
    sprintf(num, "%x", p);
    outtextxy(MaxX/4+113, MaxY/4+98, num);
}

```

```

/*****
/*      IDENTI: Entrada de nome do Programa atraves do teclado      */
/*      */
*****/

void Identi(void)
{
    int c,i,j,posi,ponto;
    int ximp,yimp;
    char dec[50],ext[6];
    char decaux[50];

    ximp= MaxX/4 + 40;
    yimp= MaxY/4 + 17;
    settextstyle(SMALL_FONT,HORIZ_DIR,4);
    setcolor(LIGHTBLUE);
    *aux= 0;
    for(i=0;i<=39;i++){
        dec[i]='';
        decaux[i]='';
    }
    i=0;
    posi= 0;
    ponto= 0;
    ext[0]= '.';
    ext[1]= 'h';
    ext[2]= 'e';
    ext[3]= 'x';
    ext[4]= '';
    while(2 == 2){
        if(i <= 39){
            c= getch();
            setfillstyle(SOLID_FILL,LIGHTGRAY);
            bar(MaxX/4+30,MaxY/4+16,MaxX*3/4-30,MaxY/4+27);
            if(c == BACK){
                if(i > 0){
                    i= i-1;
                    dec[i]='';
                    decaux[i]='';
                    posi= i;
                    if(ponto == 1){
                        if(posi == i){
                            ext[0]= '.';
                            ext[1]= 'h';
                            ext[2]= 'e';
                            ext[3]= 'x';
                            ext[4]= '';
                        }
                    }
                }
            }
        }
    }
}

```

```

if(c == ESC){
    *errsim= 0;
    *aux= 1;
    break;
}
if(c == ENTER){
    strcat(prog,ext);
    break;
}
if(c == DEL){
    setfillstyle(SOLID_FILL, LIGHTGRAY);
    bar(MaxX/4+30,MaxY/4+16,MaxX*3/4-30,MaxY/4+27);
    for(i=0;i<=39;i++){
        dec[i]='';
        decaux[i]='';
    }
    i=0;
    ponto= 0;
    posi= 0;
    c=getch();
}
if((c >= 0x61) & (c <= 0x7a)){
    dec[i]= c- 0x20;
}else{
    dec[i]= c;
}
if(c == 0x2e){
    for(j= 0;j <= 4;j++){
        ext[j]= '';
        posi= i;
        ponto= 1;
    }
}
if(c != BACK){
    decaux[i]= c;
    i= i+1;
}
sprintf(prog,"%s",decaux);
outtextxy(ximp,yimp,dec);
}else{
    setfillstyle(SOLID_FILL, LIGHTGRAY);
    bar(MaxX/4+30,MaxY/4+16,MaxX*3/4-30,MaxY/4+27);
    setcolor(LIGHTRED);
    outtextxy(ximp+20,yimp,"Muito longo");
    delay(1500);
    setfillstyle(SOLID_FILL, LIGHTGRAY);
    bar(MaxX/4+30,MaxY/4+16,MaxX*3/4-30,MaxY/4+27);
    setcolor(LIGHTBLUE);
    for(i=0;i<=39;i++){
        dec[i]='';
        decaux[i]='';
    }
}

```

```

i= 0;
ponto= 0;
posi= 0;

```

```

/*****
/*      Geracao do Grafico do Movimento      */
*****/

#define adjasp( y )      ((int)((Raspecto * (double)(y)))
#define conv( d )        ((double)(d) * PI ) / 180.0 )

void Grafm(void)
{
    char a6[15],a7[15];
    int dvg,j,x,y,co,c,escal;
    float plot[10000],valor,tempo,passo;
    int size1, *Fp1,size2, *Fp2;
    float limpg,pm,pmem;
    int ximp,yimp,t,ti,tf;
    int res,SM,i,l;
    int auxco,indseg;
    float auxpmem;
    int armaz[5],ind;

    limpg= 48.0;
    ti= 0;
    tf= 500;
    pm= (float)(tf-ti)/500;
    dvg= 30+limpg*5.73;
    ximp= 220;
    yimp= dvg+20;
    escal= 1;
    ind = 1;
    Coord(ti,tf,ximp,yimp,escal,pm,ind);

    SM= 0x5000;
    for (i=1 ; i<=1024; i++) {
        l= 0;

        while (l!=3){
            outportb(0x220,0x00);
            outportb(0x222,0x46);
            res=peekb(0x8000,SM+i-1);
            res= res & 0x00FF;
            armaz[l]= res;
            outportb(0x222,0x0F);
            outportb(0x220,0xFF);
            l= l+1;
        }
    }
}

```

```

    if((armaz[0] != armaz[1] || armaz[2]) == armaz[0]) {
        plot[i-1] = (float)(res * (48.0/256.0));
    } else {
        plot[i-1] = plot[i-2];
    }
}
Dc(plot, pm, ti, ind);
co = 0;
auxco = co;
pmem = (float)(ti * pm);
auxpmem = pmem;
escal = 1;
size1 =
imagesize(28, (dvg-plot[(int)pmem]*5.73)-5, 42, (dvg-plot[(int)pmem]
*5.73)+5);
Fp1 = malloc(size1);
size2 =
imagesize(28, (dvg-plot[(int)pmem]*5.73)-5, 42, (dvg-plot[(int)pmem]
*5.73)+5);
Fp2 = malloc(size2);
getimage(28, (dvg-plot[(int)pmem]*5.73)-5, 42, (dvg-plot[(int)pmem]
*5.73)+5, Fp1);
getimage(28, (dvg-plot[(int)pmem]*5.73)-5, 42, (dvg-plot[(int)pmem]
*5.73)+5, Fp2);
setcolor(WHITE);
line(28+co, (dvg-plot[(int)pmem]*5.73), 42+co, (dvg-plot[(int)
pmem]*5.73));
line(35+co, (dvg-plot[(int)pmem]*5.73)-5, 35+co, (dvg-plot[(int)
pmem]*5.73)+5);
indseg = 0;
while(2 == 2) {
    c = getch();
    if(c == ESC) {
        Tela();
        Telexec();
        break;
    }
    if(c == ENTER) {
        if(indseg == 0) {
            auxco = co;
            auxpmem = pmem;
            putimage(28+co, (dvg-plot[(int)pmem]*5.73)-5, Fp1, COPY_PUT
            getimage(28+co, (dvg-plot[(int)pmem]*5.73)-5, 42+co,
                (dvg-plot[(int)pmem]*5.73)+5, Fp2);
            setcolor(WHITE);
            line(28+co, (dvg-plot[(int)pmem]*5.73), 42+co,
                (dvg-plot[(int)pmem]*5.73));
            line(35+co, (dvg-plot[(int)pmem]*5.73)-5, 35+co,
                (dvg-plot[(int)pmem]*5.73)+5);
            setfillstyle(SOLID_FILL, BLACK);
            bar(MaxX-100, 40, MaxX-20, 50);
            bar(MaxX-100, MaxY-70, MaxX-20, MaxY-60);
            setcolor(WHITE);
            valor = 0;

```

```

tempo= 0;
sprintf(a6,"d: %g",valor);
strcat(a6," mm");
sprintf(a7,"dt: %g",tempo);
strcat(a7," ms");
outtextxy(MaxX-80,40,a6);
outtextxy(MaxX-80,MaxY-70,a7);
indseg= 1;
}else{
putimage(28+auxco,(dvg-plot[(int)auxpmem]*5.73)-5,Fp2,
COPY_PUT);
setfillstyle(SOLID_FILL,BLACK);
bar(MaxX-100,40,MaxX-20,50);
bar(MaxX-100,MaxY-70,MaxX-20,MaxY-60);
indseg= 0;
}
}
if(c == LETP){
escal= escal * 5;
if(escal > 125)
escal = 1;
ximp= 290;
yimp= dvg+6;
passo= (float)(escal*pm);
Mostarg(ximp,yimp,passo);
}
if(c == LETI){
ximp= 220;
yimp= dvg+20;
setfillstyle(SOLID_FILL,YELLOW);
bar(ximp-1,yimp,ximp+50,yimp+10);
setcolor(BLACK);
rectangle(ximp-1,yimp,ximp+50,yimp+10);
Entrat(ximp,yimp,&t);
setcolor(GREEN);
rectangle(ximp-1,yimp,ximp+50,yimp+10);
if((t <= 10000)&(t < tf)){
putimage(28+co,(dvg - plot[(int)pmem]*5.73)-5,Fp1,
COPY_PUT);
ti= t;
pm= (float)(tf-ti)/500;
escal= 1;
Coord(ti,tf,ximp,yimp,escal,pm,ind);
Dc(plot,pm,ti,ind);
co= 0;
pmem= ti;
getimage(28,(dvg-plot[(int)pmem]*5.73)-5,42,
(dvg-plot[(int)pmem]*5.73)+5,Fp1);
setcolor(LIGHTRED);
line(28+co,(dvg-plot[(int)pmem]*5.73),42+co,
(dvg-plot[(int)pmem]*5.73));
line(35+co,(dvg-plot[(int)pmem]*5.73)-5,35+co,
(dvg-plot[(int)pmem]*5.73)+5);
indseg= 0;
}else{

```

```

        sound(3000);
        delay(400);
        nosound();
        Mostarg(ximp,yimp,ti);
    }
}
if(c == LETF){
    ximp= 220;
    yimp= dvg+32;
    setfillstyle(SOLID_FILL,YELLOW);
    bar(ximp-1,yimp,ximp+50,yimp+10);
    setcolor(BLACK);
    rectangle(ximp-1,yimp,ximp+50,yimp+10);
    Entrat(ximp,yimp,&t);
    setcolor(GREEN);
    rectangle(ximp-1,yimp,ximp+50,yimp+10);
    if((t <= 10000)&(t > ti)){
        putimage(28+co,(dvg - plot[(int)pmem]*5.73)-5,Fp1,
        COPY_PUT);
        tf= t;
        pm= (float)(tf-ti)/500;
        escal= 1;
        Coord(ti,tf,ximp,yimp-12,escal,pm,ind);
        Dc(plot,pm,ti,ind);
        co= 0;
        pmem= ti;
        getimage(28,(dvg-plot[(int)pmem]*5.73)-5,42,
            (dvg-plot[(int)pmem]*5.73)+5,Fp1);
        setcolor(WHITE);
        line(28+co,(dvg-plot[(int)pmem]*5.73),42+co,(dvg-plot[(int)pmem]
        *5.73));
        line(35+co,(dvg-plot[(int)pmem]*5.73)-5,35+co,(dvg-plot[(int)pmem]
        *5.73)+5);
        indseg= 0;
    }else{
        sound(3000);
        delay(400);
        nosound();
        Mostarg(ximp,yimp,tf);
    }
}
if(c == LETA){
}
if(((c == SETADIREITA) & (co < 500)) ! ((co > 0) & (c ==
SETAESQUERDA))){
    if(indseg == 1){
        setcolor(WHITE);
        line(28+auxco,(dvg-plot[(int)auxpmem]*5.73),42+auxco,
            (dvg-plot[(int)auxpmem]*5.73));
        line(35+auxco,(dvg-plot[(int)auxpmem]*5.73)-5,35+auxco,
            (dvg-plot[(int)auxpmem]*5.73)+5);
    }
}

```

```

putimage(28+co,(dvg - plot[(int)pmem]*5.73)-5,Fp1,
COPY_PUT);
if((c == SETADIREITA) & (co < 500)) {
    co= co+escal;
    pmem= pmem+(pm*(float)(escal));
    if(co >= 500){
        co= 500;
        pmem= tf;
    }
    getimage(28+co,(dvg-plot[(int)pmem]*5.73)-5,42+co,
        (dvg-plot[(int)pmem]*5.73)+5,Fp1);
    setcolor(WHITE);
    line(28+co,(dvg-plot[(int)pmem]*5.73),42+co,
        (dvg-plot[(int)pmem]*5.73));
    line(35+co,(dvg-plot[(int)pmem]*5.73)-5,35+co,
        (dvg-plot[(int)pmem]*5.73)+5);
}
if((co,> 0) & (c == SETAESQUERDA)) {
    co= co-escal;
    pmem= pmem-(pm*(float)(escal));
    if(co <= 0){
        co= 0;
        pmem= ti;
    }
    getimage(28+co,(dvg-plot[(int)pmem]*5.73)-5,42+co,
        (dvg-plot[(int)pmem]*5.73)+5,Fp1);
    setcolor(WHITE);

line(28+co,(dvg-plot[(int)pmem]*5.73),42+co,(dvg-plot[(int)pmem]
*5.73));
line(35+co,(dvg-plot[(int)pmem]*5.73)-5,35+co,(dvg-plot[(int)pmem]
*5.73)+5);
}
}
if(indseg == 1){
    valor= fabs(plot[pmem]-plot[auxpmem]);
    valor= (int)(100.0 * valor);
    valor /= 100;
    tempo= fabs((co-auxco)*pm);
    tempo= tempo * 4.2;
    tempo= (1000 * tempo);
    tempo = tempo/1000;
    setfillstyle(SOLID_FILL, BLACK);
    bar(MaxX-100,40,MaxX-5,50);
    bar(MaxX-100,MaxY-70,MaxX-5,MaxY-60);
    setcolor(WHITE);
    sprintf(a6,"d: %g",valor);
    if(ind == 1)
        strcat(a6," mm");
    if(ind == 2)
        strcat(a6," N");
    sprintf(a7,"dt: %g",tempo);
    strcat(a7," ms");
}

```

```

        outtextxy(MaxX-80,40,a6);
        outtextxy(MaxX-80,MaxY-70,a7);
    }
    valor= plot[pmem];
    valor= (int)(100.0 * valor);
    valor /= 100;

    tempo= ti+ co*pm;
    tempo= tempo * 4.2;
    tempo= (1000 * tempo);
    tempo = tempo/1000;

    setfillstyle(SOLID_FILL,LIGHTGREEN);
    bar(MaxX-125,MaxY-24,MaxX-60,MaxY-1);
    setcolor(RED);
    rectangle(MaxX-125,MaxY-24,MaxX-60,MaxY-1);
    sprintf(a6,"%g",valor);
    strcat(a6," mm");
    sprintf(a7,"%g",tempo);
    strcat(a7," ms");
    setcolor(BLUE);
    outtextxy(MaxX-120,MaxY-22,a6);
    outtextxy(MaxX-120,MaxY-11,a7);
}
free(Fp1);
free(Fp2);

}
/*****
/*      COORD: Desenha coordenadas para grafico      */
*****/

void Coord(int ti,int tf,int ximp,int yimp,int escal,float pm,int ind)
{
    char a1[10],a2[10],a3[10];
    int dvj,j,auxi;
    float limpg;
    float passo;

    limpg= 48.0;
    setfillstyle(SOLID_FILL,BLACK);
    bar(0,0,MaxX,MaxY);
    setcolor(WHITE);
    settextjustify(LEFT_TEXT,TOP_TEXT);
    if(ind == 1)
        outtextxy(200,10,"Grafico de Posicao ( mm X ms )");
    if(ind == 2)
        outtextxy(200,10,"Grafico de Esforco ( N X ms )");
    setcolor(LIGHTGREEN);
    rectangle(0,0,MaxX,MaxY);
    setfillstyle(SOLID_FILL,YELLOW);
    setcolor(WHITE);
    settextjustify(LEFT_TEXT,TOP_TEXT);

```

```

settextstyle(SMALL_FONT, HORIZ_DIR, 4);
line(35, 25, MaxX-80, 25);
dvg= 30+limpg*5.73;
line(35, 30, 35, 305);
bar(0, 310, MaxX, MaxY);
if(ind == 1){
    sprintf(a1, "%g", limpg);
    outtextxy(10, 25, a1);
    sprintf(a2, "%d", 0);
    outtextxy(10, 300, a2);
    line(30, dvg, MaxX-60, dvg);
    for( j=1 ; j<=9; ++j ){
        auxi = 5*j;
        sprintf(a3, "%d", auxi);
        outtextxy(10, ((dvg-5) - 5*j*5.73), a3);
    }
}
if(ind == 2){
    sprintf(a2, "%d", 0);
    outtextxy(10, 300, a2);
    line(30, dvg, MaxX-60, dvg);
    for( j=1 ; j<=9; ++j ){
        auxi = 21*j;
        sprintf(a3, "%d", auxi);
        outtextxy(10, ((dvg-5) - 30.55*j), a3);
    }
}
setcolor(BLUE);
line(5, dvg+18, MaxX-10, dvg+18);
outtextxy(35, dvg+20, "Junta");
setfillstyle(SOLID_FILL, BLACK);
bar(5, dvg+30, 100, dvg+48);
setcolor(RED);
outtextxy(25, dvg+35, "- Garra -");
setcolor(MAGENTA);
outtextxy(250, dvg+6, "P");
outtextxy(140, dvg+20, "I");
outtextxy(140, dvg+32, "F");
outtextxy(320, dvg+20, "A");
outtextxy(320, dvg+32, "ESC - Voltar Execucao");
setcolor(BLUE);
outtextxy(250, dvg+6, "asso:");
outtextxy(140, dvg+20, "Ponto nicial:");
outtextxy(140, dvg+32, "Ponto inal:");
outtextxy(320, dvg+20, "juste para imprimir");
Mostarg(ximp, yimp, ti);
yimp= yimp+12;
Mostarg(ximp, yimp, tf);
ximp= 290;
yimp= dvg+6;
passo= (float)(escal*pm);
Mostarg(ximp, yimp, passo);

```

```

/*****
/*      DC: Desenha curva
*****/

void Dc(float plot[10000],float pm,int ti,int ind)
{
    int dvg,j,x,y;
    float indic,limpg;

    limpg= 48.0;
    dvg= 30+limpg*5.73;

    for( j=0 ; j<=500; ++j ){
        indic= (float)(ti+(j*pm));
        x= 35+j;
        if(ind == 1)
            y= dvg - plot[(int)indic]*5.73;
        if(ind == 2)
            y= dvg - plot[(int)indic]*1.4550;

        putpixel(x,y,LIGHTBLUE);

    }
    sound(100);
    delay(20);
    nosound();
}

/*****
/*      Telexec: Desenha Telinha com Menu da Execucao
*****/

void Telexec(void)
{
    setcolor(WHITE);
    setfillstyle(SOLID_FILL,RED);
    bar(MaxX/4+150,MaxY/4+10,MaxX*3/4-10,MaxY*3/4-50);
    rectangle(MaxX/4+150,MaxY/4+10,MaxX*3/4-10,MaxY*3/4-50);
    setttextjustify(CENTER_TEXT,TOP_TEXT);
    setttextstyle(DEFAULT_FONT,HORIZ_DIR,1);
    outtextxy(MaxX/2+70,MaxY/4+20,"Execucao");
    setcolor(WHITE);
    setttextjustify(LEFT_TEXT,TOP_TEXT);
    setttextstyle(SMALL_FONT,HORIZ_DIR,4);
    outtextxy(MaxX/4+155,MaxY/4+40,"P");
    outtextxy(MaxX/4+155,MaxY/4+60,"A");
    outtextxy(MaxX/4+155,MaxY/4+80,"Grafico");
    outtextxy(MaxX/4+155,MaxY/4+100,"Grafico");
    setcolor(YELLOW);
    outtextxy(MaxX/4+155,MaxY/4+40,"osicao desejada");
    outtextxy(MaxX/4+155,MaxY/4+60,"mostrar");
    outtextxy(MaxX/4+155,MaxY/4+80,"1");
    outtextxy(MaxX/4+155,MaxY/4+100,"2");
}

```

```

    setcolor(LIGHTGREEN);
    circle(450,110,6);
}
/*****
/*      Mostarg: Coloca no rodape o argumento desejado      */
*****/

void Mostarg(int ximp,int yimp,float arg)
{
    char imp[10];
    setfillstyle(SOLID_FILL,YELLOW);
    bar(ximp-1,yimp,ximp+50,yimp+10);
    sprintf(imp,"%g",arg);
    setcolor(WHITE);
    outtextxy(ximp+1,yimp,imp);
}

```

```

/*                                     Campinas,19/07/91      */
/*
/*
/* Este Programa Faz Parte do Pacote para Controlar Trajetorias */
/* de Robos-Manipuladores Mecanicos. Documento                */
/* Rotina Garra 3                                              */
/* Finalidade: Testar controle PID da garra atraves do PC     */

#include <dos.h>
#include <dir.h>
#include <math.h>
#include <conio.h>
#include <stdio.h>          /* Inclui as bibliotecas necessarias */
#include <stdlib.h>
#include <stdarg.h>
#include <graphics.h>
#include <process.h>
#include <string.h>
#include "gc.h"

extern int      GraphDriver;          /* Tipo de driver grafico */
extern int      GraphMode;            /* Valor do modo grafico */
extern double   Raspecto;             /* Razao de aspecto dos pixels na tela */
extern int      MaxX, MaxY;           /* Maxima Resolucao da tela */
extern int      MaxColors;            /* Maximo de cores avaliaveis */
extern int      ErrorCode;            /* Codigo de erros graficos */
extern struct palettetype palette;    /* Usado para ler informacoes
do palette */

extern int      Grav[800];
extern char     prog[60];
extern int      *aux,*aux1,*errsim;
extern int      indpgr;

struct PTS {
    int x, y;
};      /* Estrutura para segurar pontos de vertice

```

```

/*****
/*      Geracao do Grafico do Esforco      */
*****/

#define adjasp( y )      ((int)(Raspecto * (double)(y)))
#define conv( d )        ((double)(d) * PI ) / 180.0 )

void Grafforc(void)
{
    char a6[15],a7[15];
    int dvg,j,x,y,co,c,escal;
    float plot[10000],valor,tempo,passo;
    int size1, *Fp1,size2, *Fp2;
    float limpg,pm,pmem;
    int ximp,yimp,t,ti,tf;
    int res,SM,i,l;
    int auxco,indseg;
    float auxpmem;
    int armaz[5],ind;

    limpg= 189.0;
    ti= 0;
    tf= 500;
    pm= (float)(tf-ti)/500;
    dvg= 30+limpg*1.4550;
    ximp= 220;
    yimp= dvg+20;
    escal= 1;
    ind= 2;
    Coord(ti,tf,ximp,yimp,escal,pm,ind);
    SM= 0x5000;
    for (i=1 ; i<=1024; i++) {
        l= 0;
        while (l!=3){
            outportb(0x220,0x00);
            outportb(0x222,0x46);
            res=peekb(0x8000,SM+i-1);
            res= res & 0x00FF;
            armaz[l]= res;
            outportb(0x222,0x0F);
            outportb(0x220,0xFF);
            l= l+1;
        }
        if((armaz[0] != armaz[1] != armaz[2]) == armaz[0]){
            plot[i-1]= (float) ((res - 128.0) * (189.0/128.0));
        }else{
            plot[i-1]= plot[i-2];
        }
        if(res < 128.0)
            plot[i-1] = 0.0;
    }
    Dc(plot,pm,ti,ind);
}

```

```

co= 0;
auxco= co;
pmem= (float)(ti*pm);
auxpmem= pmem;
escal= 1;
size1 =
imageSize(28,(dvg-plot[(int)pmem]*1.4550)-5,42,(dvg-plot[(int)pmem]
*1.4550)+5);
Fp1 = malloc(size1);
size2 =
imageSize(28,(dvg-plot[(int)pmem]*1.4550)-5,42,(dvg-plot[(int)pmem]
*1.4550)+5);
Fp2 = malloc(size2);
getImage(28,(dvg-plot[(int)pmem]*1.4550)-5,42,(dvg-plot[(int)pmem]
*1.4550)+5,Fp1);
getImage(28,(dvg-plot[(int)pmem]*1.4550)-5,42,(dvg-plot[(int)pmem]
*1.4550)+5,Fp2);
setColor(WHITE);
line(28+co,(dvg-plot[(int)pmem]*1.4550),42+co,(dvg-plot[(int)pmem]
*1.4550));
line(35+co,(dvg-plot[(int)pmem]*1.4550)-5,35+co,(dvg-plot[(int)pmem]
*1.4550)+5);

indseg= 0;
while(2 == 2){
    c= getch();
    if(c == ESC){
        Tela();
        Telexec();
        break;
    }
    if(c == ENTER){
        if(indseg == 0){
            auxco= co;
            auxpmem= pmem;
            putImage(28+co,(dvg - plot[(int)pmem]*1.4550)-5,Fp1,
COPY_PUT);
            getImage(28+co,(dvg-plot[(int)pmem]*1.4550)-5,42+co,
(dvg-plot[(int)pmem]*1.4550)+5,Fp2);
            setColor(WHITE);
            line(28+co,(dvg-plot[(int)pmem]*1.4550),42+co,
(dvg-plot[(int)pmem]*1.4550));
            line(35+co,(dvg-plot[(int)pmem]*1.4550)-5,35+co,
(dvg-plot[(int)pmem]*1.4550)+5);

            setfillstyle(SOLID_FILL,BLACK);
            bar(MaxX-100,40,MaxX-20,50);
            bar(MaxX-100,MaxY-70,MaxX-20,MaxY-60);
            setColor(WHITE);
            valor= 0;
            tempo= 0;
            sprintf(a6,"d: %g",valor);

```

```

    strcat(a6," N");
    sprintf(a7,"dt: %g",tempo);
    strcat(a7," ms");
    outtextxy(MaxX-80,40,a6);
    outtextxy(MaxX-80,MaxY-70,a7);
    indseg= 1;
} else{
    putimage(28+auxco,(dvg-plot[(int)auxpmem]*1.4550)-5,Fp2,
    COPY_PUT);
    setfillstyle(SOLID_FILL, BLACK);
    bar(MaxX-100,40,MaxX-20,50);
    bar(MaxX-100,MaxY-70,MaxX-20,MaxY-60);
    indseg= 0;
}
}
if(c == LETP){
    escal= escal * 5;
    if(escal > 125)
        escal = 1;
    ximp= 290;
    yimp= dvg+6;
    passo= (float)(escal*pm);
    Mostarg(ximp,yimp,passo);
}
if(c == LETI){
    ximp= 220;
    yimp= dvg+20;
    setfillstyle(SOLID_FILL, YELLOW);
    bar(ximp-1,yimp,ximp+50,yimp+10);
    setcolor(BLACK);
    rectangle(ximp-1,yimp,ximp+50,yimp+10);
    Entrat(ximp,yimp,&t);
    setcolor(GREEN);
    rectangle(ximp-1,yimp,ximp+50,yimp+10);
    if((t <= 10000)&(t < tf)){
        putimage(28+co,(dvg - plot[(int)pmem]*1.4550)-5,Fp1,
        COPY_PUT);
        ti= t;
        pm= (float)(tf-ti)/500;
        escal= 1;
        Coord(ti,tf,ximp,yimp,escal,pm,ind);
        Dc(plot,pm,ti,ind);
        co= 0;
        pmem= ti;
        getimage(28,(dvg-plot[(int)pmem]*1.4550)-5,42,
        (dvg-plot[(int)pmem]*1.4550)+5,Fp1);
        setcolor(LIGHTRED);
        line(28+co,(dvg-plot[(int)pmem]*1.4550),42+co,
        (dvg-plot[(int)pmem]*1.4550));
        line(35+co,(dvg-plot[(int)pmem]*1.4550)-5,35+co,
        (dvg-plot[(int)pmem]*1.4550)+5);
        indseg= 0;
    } else{

```

```

        sound(100);
        delay(20);
        nosound();
        Mostarg(ximp,yimp,ti);
    }
}
if(c == LETF){
    ximp= 220;
    yimp= dvlg+32;
    setfillstyle(SOLID_FILL,YELLOW);
    bar(ximp-1,yimp,ximp+50,yimp+10);
    setcolor(BLACK);
    rectangle(ximp-1,yimp,ximp+50,yimp+10);
    Entrat(ximp,yimp,&t);
    setcolor(GREEN);
    rectangle(ximp-1,yimp,ximp+50,yimp+10);
    if((t <= 10000)&(t > ti)){
        putimage(28+co,(dvlg - plot[(int)pmem]*1.4550)-5,Fp1,
        COPY_PUT);
        tf= t;
        pm= (float)(tf-ti)/500;
        escal= 1;
        Coord(ti,tf,ximp,yimp-12,escal,pm,ind);
        Dc(plot,pm,ti,ind);
        co= 0;
        pmem= ti;
        getimage(28,(dvlg-plot[(int)pmem]*1.4550)-5,42,
            (dvlg-plot[(int)pmem]*1.4550)+5,Fp1);
        setcolor(WHITE);

line(28+co,(dvlg-plot[(int)pmem]*1.4550),42+co,(dvlg-plot[(int)pmem]
*1.4550));
line(35+co,(dvlg-plot[(int)pmem]*1.4550)-5,35+co,(dvlg-plot[(int)pmem]
*1.4550)+5);
        indseg= 0;
    }else{
        sound(100);
        delay(20);
        nosound();
        Mostarg(ximp,yimp,tf);
    }
}
if(c == LETA){
}
if(((c == SETADIREITA) & (co < 500)) ; ((co > 0) & (c ==
SETAESQUERDA))){
    if(indseg == 1){
        setcolor(WHITE);
        line(28+auxco,(dvlg-plot[(int)auxpmem]*1.4550),42+auxco,
            (dvlg-plot[(int)auxpmem]*1.4550));
        line(35+auxco,(dvlg-plot[(int)auxpmem]*1.4550)-5,35+auxco,
            (dvlg-plot[(int)auxpmem]*1.4550)+5);
    }
}

```

```

putimage(28+co,(dvg - plot[(int)pmem]*1.4550)-5,Fp1,
COPY_PUT);
if((c == SETADIREITA) & (co < 500))<
    co= co+escal;
    pmem= pmem+(pm*(float)(escal));
    if(co >= 500)<
        co= 500;
        pmem= tf;
    >
    getimage(28+co,(dvg-plot[(int)pmem]*1.4550)-5,42+co,
        (dvg-plot[(int)pmem]*1.4550)+5,Fp1);
    setcolor(WHITE);
    line(28+co,(dvg-plot[(int)pmem]*1.4550),42+co,
        (dvg-plot[(int)pmem]*1.4550));
    line(35+co,(dvg-plot[(int)pmem]*1.4550)-5,35+co,
        (dvg-plot[(int)pmem]*1.4550)+5);
>
if((co > 0) & (c == SETAESQUERDA))<
    co= co-escal;
    pmem= pmem-(pm*(float)(escal));
    if(co <= 0)<
        co= 0;
        pmem= ti;
    >
    getimage(28+co,(dvg-plot[(int)pmem]*1.4550)-5,42+co,
        (dvg-plot[(int)pmem]*1.4550)+5,Fp1);
    setcolor(WHITE);

line(28+co,(dvg-plot[(int)pmem]*1.4550),42+co,(dvg-plot[(int)pmem]
*1.4550));
line(35+co,(dvg-plot[(int)pmem]*1.4550)-5,35+co,(dvg-plot[(int)pmem]
*1.4550)+5);
>
>
if(indseg == 1)<
    valor= fabs(plot[pmem]-plot[auxpmem]);
    valor= (int)(100.0 * valor);
    valor /= 100;

    tempo= fabs((co-auxco)*pm);
    tempo= tempo * 4.2;
    tempo= (1000 * tempo);
    tempo = tempo/1000;
    setfillstyle(SOLID_FILL,BLACK);
    bar(MaxX-100,40,MaxX-5,50);
    bar(MaxX-100,MaxY-70,MaxX-5,MaxY-60);
    setcolor(WHITE);
    sprintf(a6,"d: %g",valor);
    strcat(a6," N");
    sprintf(a7,"dt: %g",tempo);
    strcat(a7," ms");
    outtextxy(MaxX-80,40,a6);
    outtextxy(MaxX-80,MaxY-70,a7);
>

```

```

valor= plot[pmem];
valor= (int)(100.0 * valor);
valor /= 100;

tempo= ti+ co*pm;
tempo= tempo * 4.2;
tempo= (1000 * tempo);
tempo = tempo/1000;

setfillstyle(SOLID_FILL, LIGHTGREEN);
bar(MaxX-125, MaxY-24, MaxX-60, MaxY-1);
setcolor(RED);
rectangle(MaxX-125, MaxY-24, MaxX-60, MaxY-1);
sprintf(a6, "%g", valor);
strcat(a6, " N");
sprintf(a7, "%g", tempo);
strcat(a7, " ms");
setcolor(BLUE);
outtextxy(MaxX-120, MaxY-22, a6);
outtextxy(MaxX-120, MaxY-11, a7);
}
free(Fp1);
free(Fp2);

```

PROGRAMAS DESENVOLVIDOS EM ASSEMBLER

```

*****
*
*      MONITOR PCDDII
*
*      23/03/91
*
*****

```

\$ALLPUBLIC

```

      DEFSEG DEFCODE, START=0
      SEG     DEFCODE
STTOP: EQU     7FFFH

      DEFSEG IOSEG, START=0, CLASS=IOSPACE
      SEG     IOSEG
      ORG     00H
PORT_A: DS     1      ; Endereco da porta digital A de E/S
PORT_B: DS     1      ; Endereco da porta digital B de E/S
PORT_C: DS     1      ; Endereco do porta digital C de E/S
CONTE: DS     1      ; Endereco da Palavra de controle da PIA
CDASUP: DS     1      ; Bits A9 e A8 do D/A
CDAINF: DS     1      ; Bits A0 a A7 do D/A
EXPAN1: DS     1      ; Expansao
EXPAN2: DS     1      ; Expansao
CANADC: DS     1      ; Canais de entrada A/D
INICON: DS     1      ; Inicio de conversao
LADINF: DS     1      ; Byte inferior do A/D
LADSUP: DS     1      ; Byte superior do A/D
CANTE0: DS     1      ; Endereco do canal de temporizacao 0
CANTE1: DS     1      ; Endereco do canal de temporizacao 0
CANTE2: DS     1      ; Endereco do canal de temporizacao 0
CONTEP: DS     1      ; Endereco da Palavra de controle do 8253
ICW1TP: DS     1      ; Endereca da Palavra de controle 1 do 8259
ICW2TP: DS     1      ; Endereca da Palavra de controle 2 do 8259

      ORG     1CH
INTACP: DS     1      ; Simulacao do sinal de INTA para o 8259

```

*****Inicializacao de variaveis*****

```

INIRAM: EQU     4000H      ;Endereco inicial da RAM
FIMRAM: EQU     7FFFH      ;Endereco final da RAM
CONPIA: EQU     80H        ;Seta P_A=E & P_B=S & P_C=S
TEMPO1: EQU     01H        ;Variavel do DELAY
TEMPO2: EQU     0FFFH      ;Variavel do DELAY
PCONTO: EQU     34H        ;Palavra de controle do canal 0
PCONT1: EQU     74H        ;Palavra de controle do canal 1
PCONT2: EQU     0B4H       ;Palavra de controle do canal 2
ITAB1: EQU     5000H       ;Inicio da tabela 1
ITAB2: EQU     5800h       ;Inicio da tabela 2

```

*****CONT DE INTERRUPCAO *****

```
ICW1CI: EQU      13H      ;
ICW2CI: EQU      70H      ;
ICW4CI: EQU      0FH      ;
CAN0AD: EQU      00H      ; Canal Analogico 0
CAN1AD: EQU      01H      ; Canal Analogico 1
CAN2AD: EQU      02H      ; Canal Analogico 2
CAN3AD: EQU      03H      ; Canal Analogico 3
CAN4AD: EQU      04H      ; Canal Analogico 4
CAN5AD: EQU      05H      ; Canal Analogico 5
CAN6AD: EQU      06H      ; Canal Analogico 6
CAN7AD: EQU      07H      ; Canal Analogico 7
```

```
DEFSEG ZEROCODE,START=0
SEG      ZEROCODE
```

*****Endereco de Reset *****

```
ORG      0000H
IM        1
DI
LD        HL,STTOP ;Inicializacao do Stack Pointer
LD        SP,HL
LD        HL,ITAB1
LD        (PONT1),HL
LD        HL,ITAB2
LD        (PONT2),HL
LD        A,00H
LD        (EW),A
LD        (HINT),A
LD        (EW1),A
JP        INICIO ;Rotina de Inicializacao
```

*****Endereco Inicial de Interrupcao*****

```
ORG      0038H
DI
PUSH      AF
PUSH      BC
PUSH      DE
PUSH      HL
PUSH      IX
PUSH      IY

LD        A,00H
OUT       (INTACP),A
IN        A,(INTACP)
EI
```

```

VER0:  CP      70H
        JP      NZ, VER1
        CALL    4000H
        JP      NFIM

VER1:  CP      71H
        JP      NZ, VER2
        JP      NFIM

VER2:  CP      72H
        JP      NZ, VER3
        JP      NFIM

VER3:  CP      73H
        JP      NZ, VER4
        JP      NFIM

VER4:  CP      74H
        JP      NZ, VER5
        JP      NFIM

VER5:  CP      75H
        JP      NZ, VER6
        JP      NFIM

VER6:  CP      76H                ;Leitura do ADC
        JP      NZ, VER7
        IN      A, (LADINF)
        LD      (ADCINF), A
        IN      A, (LADSUP)
        LD      (ADCSUP), A
        JP      NFIM

VER7:  CP      77H
        JP      NZ, NFIM
        JP      NFIM

NFIM:

        POP     IY
        POP     IX
        POP     HL
        POP     DE
        POP     BC
        POP     AF
        RETI

        ORG     0500H

```

INICIO:

```
*****
*
*          ROTINA DE INICIALIZACAO
*
*****
```

***** Inicializacao Da PIA *****

```
LD      A,CONPIA
OUT     (CONTES),A
LD      A,80H
OUT     (PORT_A),A
```

***** Inicializacao do Cont de Interrupcao *****

```
LD      A,ICW1CI
OUT     (ICW1TP),A

LD      A,ICW2CI
OUT     (ICW2TP),A

LD      A,ICW4CI
OUT     (ICW2TP),A

LD      A,OBEH
OUT     (ICW2TP),A
```

***** Inicializacao dos Temporizadores *****

CANALO:

```
LD      A,PCONT0
OUT     (CONTEP),A
LD      A,080H
OUT     (CANTE0),A
LD      A,20H
OUT     (CANTE0),A
```

CANAL1:

```
LD      A,PCONT1
OUT     (CONTEP),A
LD      A,00H
OUT     (CANTE1),A
LD      A,00H
OUT     (CANTE1),A
```

CANAL2:

```

LD      A,PCONT2
OUT     (CONTEP),A
LD      A,00H
OUT     (CANTE2),A
LD      A,00H
OUT     (CANTE2),A

```

***** PROGRAMA PRINCIPAL *****

```

TEST:   LD      A,CHINTD
        CP      OFFH
        JP      Z,TINT
        JP      TEST
TINT:   EI
CONT:   LD      A,CHINTD
        CP      00H
        JP      NZ,CONT
        DI
        LD      A,80H
        OUT     (PORT_AD),A
        JP      TEST

```

```

*****
*
*          ROTINA   DELAY
*
*****

```

```

DELAY:  PUSH     BC
        PUSH     DE
        LD       B,TEMPO1      ;PODEM SER PASSADAS POR
        LD       DE,TEMPO2     ;SOFTWARE
DELAY1:  DEC      DE
        LD       A,E
        OR       D
        JR      NZ,DELAY1

        DEC      B
        LD       A,B
        OR       B
        JR      NZ,DELAY1

        POP      DE
        POP      BC

        RET

```

*****DADOS*****

```
      ORG      74FFH
ADCINF: DS      1      ; BYTE INFERIOR DO ADC
ADCSUP: DS      1      ; BYTE SUPERIOR DO ADC
MFLAGS: DS      1      ; BYTE DE TESTE INTERNO DE MEMORIA RAM
      ORG      7505H
PROGRI: DS      1      ; Variavel de Saida Para PIA
HINT:   DS      1
      ORG      7617H
EW:     DS      1      ; 7619H - HABILITACAO DE ESCRITA DE NA MEMORIA
PONT1:  DS      2      ; 761AH - PONTEIRO DA TABELA DE POSICOES 1
PONT2:  DS      2      ; 761CH - PONTEIRO DA TABELA DE PODICOES 2
EW1:    DS      1      ; 761EH -
```

*****PILHA*****

```
      ORG      7D00H
STACK: DS      255
```

END

\$ALLPUBLIC

```
*****
*
* PROGRAMA DE CONTROLE PROPORCIONAL DERIVATIVO PARA A PCDDII *
* ELABORADO POR MARCUS-SAKAHI-MADRID *
* DATA 22-06-91 *
*****
```

```
*****
*
* FUNCAO REGULADOR PROPORCIONAL DERIVATIVO *
*
*****
```

DEFSEG IOSEG, START=00H, CLASS=IOSPACE

SEG IOSEG

ORG 00H

```
PORT_A: DS 1 ; Endereco da porta digital A de E/S
PORT_B: DS 1 ; Endereco da porta digital B de E/S
PORT_C: DS 1 ; Endereco do porta digital C de E/S
CONTEC: DS 1 ; Endereco da Palavra de controle da PIA
CDAINF: DS 1 ; Bits A9 e A8 do D/A
CDASUP: DS 1 ; Bits A0 a A7 do D/A
EXPAN1: DS 1 ; Expansao
EXPAN2: DS 1 ; Expansao
CANADC: DS 1 ; Canais de entrada A/D
INICON: DS 1 ; Inicio de conversao
LADINF: DS 1 ; Byte inferior do A/D
LADSUP: DS 1 ; Byte superior do A/D
CANTE0: DS 1 ; Endereco do canal de temporizacao 0
CANTE1: DS 1 ; Endereco do canal de temporizacao 0
CANTE2: DS 1 ; Endereco do canal de temporizacao 0
CONTEP: DS 1 ; Endereco da Palavra de controle do 8253
ICW1TP: DS 1 ; Endereco da Palavra de controle 1 do 8259
ICW2TP: DS 1 ; Endereco da Palavra de controle 2 do 8259
```

ORG 1CH

```
INTACP: DS 1 ; Simulacao do sinal de INTA para o 8259
```

DEFSEG ZEROCODE, START=0

SEG ZEROCODE

DEFSEG MAINCODE, START=00H

SEG MAINCODE

ORG 4000H

```
CAN0AD: EQU 00H ; Canal Analogico 0
CAN1AD: EQU 01H ; Canal Analogico 1
CAN2AD: EQU 02H ; Canal Analogico 2
CAN3AD: EQU 03H ; Canal Analogico 3
CAN4AD: EQU 04H ; Canal Analogico 4
CAN5AD: EQU 05H ; Canal Analogico 5
```

```

CAN6AD: EQU      06H      ; Canal Analogico 6
CAN7AD: EQU      07H      ; Canal Analogico 7
INITAB: EQU      5000H
FIMTAB: EQU      57FFH
INITA2: EQU      5800H
FIMTA2: EQU      5FFFH

```

```

*
*          MODULO DE AQUISICAO DE DADOS
*
*

```

```

;      Modulo AD;
;
;      Entrada portan,address;
;      Saida   Reg.A;
;
;
LD      A,CANOAD
OUT     (CANADC),A
OUT     (INICON),A      ; Inicio de Conversao
HALT
LD      A,(EWD)
CP      00
LD      A,(7500H)
JP      Z,OFFSET
LD      HL,(PONT1)
LD      (HL),A
INC     HL
LD      B,H
LD      C,L
LD      DE,FIMTAB
SBC     HL,DE
JP      NZ,TAB
LD      B,A
LD      A,00H
LD      (EWD),A
LD      HL,INITAB
LD      (PONT1),HL
LD      A,B
JP      OFFSET
TAB:    LD      H,B
LD      L,C
LD      (PONT1),HL

```

```

;      Fim do Modulo AD;
;
;
;

```

```

*****
*
*      Modulo Conv. Offset-Bin. p/ Sinal+Modulo
*
*****

```

```

;      Modulo Off_Sinal;
;      Entrada Reg. A;
;      Saida   Sinal, Modulo;

OFFSET: BIT      7,A          ; TESTA SE POSITIVO OU NEGATIVO
        LD       B,00H        ;
        JP       Z,NEG1       ; SE NEGATIVO VAI PARA NEG
        RES      7,A          ;
        JP       CARR         ;
NEG1:    LD       B,OFFH       ;
        NEG      ;
        JP       Z,ZERO       ;
        RES      7,A          ;
        JP       CARR         ;
ZERO:    SET      7,A          ;
CARR:    LD       (modulo),A    ; CARREGA MODULO DO DADO
        LD       A,B          ;
        LD       (sinal),A     ; CARREGA SINAL DO DADO

;      Fim do Modulo Off_Sinal;

```

```

*****
*
*      Modulo de Calculo do Erro
*
*****

```

```

;      Modulo Erro;
;      Entrada sinal, modulo, sref, mref;
;      Saida   serro0, merro0;

        .
LD       D,00H          ;
LD       B,A            ; CARREGA EM B O SINAL
LD       A,(sref)       ;
XOR      B              ; TESTA SE SINAIS IGUAIS OU DIFERENTES
JP       NZ,CAS03       ; SE DIFERENTES VAI PARA CAS03
LD       A,OFFH         ;
ADD      A,B            ; TESTA SE SINAL E POSITIVO OU NEGATIVO
JP       C,CAS01        ; SE NEGATIVO VAI PARA CAS01
LD       A,(modulo)     ; ARRANJO DOS DADOS P/ FAZER MREF-MODULO
LD       B,A            ;
LD       A,(mref)       ;
JP       CAS04          ;

```

```

CAS01: LD      A,(mref)      ;ARRANJO DOS DADOS P/ FAZER MODULO-MREF
      LD      B,A          ;
      LD      A,(modulo)    ;
CAS04: SUB      B          ;
      LD      D,00         ;
      BIT     7,A          ;TESTA SE RESULTADO E POSITIVO OU NEGATIVO
      JP      Z,FIM        ;SE POSITIVO VAI PARA FIM
      LD      D,OFFH       ;SENAO FAZER COMPLEMENTO
      NEG     ;
      JP      FIM          ;
CAS03: LD      A,OFFH       ;SE SINAIS DIFERENTES FAZER MODULO+SREF
      ADD     A,B          ;TESTAR SE RESULTADO E POSITIVO OU NEG/.
      JP      C,CAS02      ;SE POSITIVO VAI PARA CAS02
      LD      D,OFFH       ;
CAS02: LD      A,(modulo)    ;
      LD      B,A          ;
      LD      A,(mref)     ;
      ADD     B            ;
      JP      NC,FIM       ;SE CARRY FAZER SATURACAO
      LD      A,OFFH       ;
FIM:   LD      (merro0),A    ;
      LD      A,D          ;
      LD      (serro0),A    ;
      ;      Fim do Modulo Erro;

```

```

*****
*                                     *
*      Modulo do Regulador PD        *
*                                     *
*****

```

```

;      Modulo P;
;      Entrada serro0, merro0, s0, mq0;
;      Saida  ssaida, msaida;

```

```

PD:   LD      B,A          ;PREPARACAO DOS DADOS P/ FAZER SQ0*ERRRO
      LD      A,(merro0)    ;
      LD      E,A          ;
      LD      A,(mq0)       ;
      LD      D,A          ;
      LD      A,(sq0)       ;
      CALL    MULT          ;
      LD      (maxc0),HL    ;COLOCA MODULO DO RESULTADO EM MAXCO
      LD      (saxc0),A     ;COLOCA SINAL DO RESULTADO EM SAXCO
      LD      A,(serro1)    ;PREPARACAO DOS DADOS P/ FAZER SQ1*ERR01
      LD      B,A          ;
      LD      A,(merro1)    ;
      LD      E,A          ;
      LD      A,(mq1)       ;
      LD      D,A          ;
      LD      A,(sq1)       ;

```

```

CALL    MULT          ;
LD      D,H           ;COLOCA MODULO DO RESULTADO EM DE
LD      E,L           ;
LD      B,A           ;COLOCA SINAL DO RESULTADO EM B
LD      HL,(maxc0)    ;PREPARACAO DOS DADOS,P/ FAZER AXC2+AXC1
LD      A,(saxc0)     ;
CALL    SOMA          ;
LD      (ssaida),A    ;FORMATACAO DO RESULTADO FINAL EM OITO BITS
LD      A,00H         ;
OR      H             ;
JP      Z,PAUX4       ;
LD      A,OFFH        ;
JP      PAUX3         ;
PAUX4:  BIT           7,L ;
JP      Z,PAUX2       ;
LD      A,OFFH        ;
JP      PAUX3         ;
PAUX2:  LD            A,L ;
PAUX3:  LD            (msaida),A ;

```

```

;      fim do Modulo PD
;
;

```

```

*****
*
*      MODULO DE AQUISICAO DE DADOS      *
*      DE FORCA                          *
*****

```

```

;      Modulo AD;
;      Entrada portan,address;
;      Saida   Reg. A;
LD      A,CAN1AD
OUT      (CANADC),A
OUT      (INICON),A      ; Inicio de Conversao
HALT
LD      A,(EW1)
CP      00H
LD      A,(7500H)
JP      Z,OFFSE2
LD      HL,(PONT2)
LD      (HL),A
INC     HL
LD      B,H
LD      C,L
LD      DE,FIMTA2
SBC     HL,DE
JP      NZ,TAB2
LD      B,A
LD      A,00H

```

```

LD      CEW1),A
LD      HL,INITA2
LD      (PONT2),HL
LD      A,B
JP      OFFSE2
TAB2:   LD      H,B
LD      L,C
LD      (PONT2),HL

```

```

;      Fim do Modulo AD;
;
;

```

```

*****
*
*      Modulo Conv. Offset-Bin. p/ Sinal+Modulo
*
*****

```

```

;      Modulo Off_Sinal;
;      Entrada Reg. A;
;      Saida   Sinal, Modulo;

```

```

OFFSE2: BIT      7,A          ;TESTA SE POSITIVO OU NEGATIVO
LD      B,00H
JP      Z,NEG2          ;SE NEGATIVO VAI PARA NEG
RES      7,A
JP      CARR2
NEG2:   LD      B,OFFH
NEG
JP      Z,ZERO2
RES      7,A
JP      CARR2
ZERO2:  SET      7,A
CARR2:  LD      (modulo),A    ;CARREGA MODULO DO DADO
LD      A,B
CPL
LD      (sinal),A          ;CARREGA SINAL DO DADO

```

```

;      Fim do Modulo Off_Sinal;

```

```

*****
*
*      Modulo de Calculo do Erro
*
*****

```

```

;      Modulo Erro;
;      Entrada sinal, modulo, sref, mref;
;      Saida   serro0, merro0;

```

```

LD      D,00H      ;
LD      B,A        ;CARREGA EM B O SINAL
LD      A,(ssaida) ;
XOR     B          ;TESTA SE SINAIS IGUAIS OU DIFERENTES
JP      NZ,CAS032  ;SE DIFERENTES VAI PARA CAS03
LD      A,OFFH     ;
ADD     A,B        ;TESTA SE SINAL E POSITIVO OU NEGATIVO
JP      C,CAS012   ;SE NEGATIVO VAI PARA CAS01
LD      A,(modulo) ;ARRANJO DOS DADOS P/ FAZER MREF-MODULO
LD      B,A        ;
LD      A,(msaida) ;
JP      CAS042     ;
CAS012: LD      A,(msaida) ;ARRANJO DOS DADOS P/ FAZER MODULO-MREF
LD      B,A        ;
LD      A,(modulo) ;
CAS042: SUB     B   ;
LD      D,00      ;
BIT     7,A        ;TESTA SE RESULTADO E POSITIVO OU NEGATIVO
JP      Z,FIM2     ;SE POSITIVO VAI PARA FIM
LD      D,OFFH     ;SENAO FAZER COMPLEMENTO
NEG     ;
JP      FIM2       ;
CAS032: LD      A,OFFH ;SE SINAIS DIFERENTES FAZER MODULO+SREF
ADD     A,B        ;TESTAR SE RESULTADO E POSITIVO OU NEG/.
JP      C,CAS022   ;SE POSITIVO VAI PARA CAS02
LD      D,OFFH     ;
CAS022: LD      A,(modulo) ;
LD      B,A        ;
LD      A,(msaida) ;
ADD     B          ;
JP      NC,FIM2    ;SE CARRY FAZER SATURACAO
LD      A,OFFH     ;
FIM2:   LD      (msaida),A ;
LD      A,D        ;
LD      (ssaida),A ;
;      Fim do Modulo Erro;

```

```

*****
*                                     *
*      Modulo Conv. Sinal+Modulo p/ Offset      *
*                                     *
*****

```

```

;      Modulo Sinal;
;      Entrada ssaida, msaida;
;      Saida  valor;

LD      A,(msaida) ;
LD      H,A        ;
LD      A,(ssaida) ;
RRC     A          ;

```

```

      JP      C,MUDA      ;
      BIT     7,H         ;
      JP      Z,M1        ;
      LD      A,07FH      ;
M1:   SET     7,H         ;
      LD      A,H         ;
      JP      FMS         ;
MUDA: BIT     7,H         ;
      JP      Z,M2        ;
      LD      H,80H       ;
M2:   LD      A,H         ;
      NEG                     ;
      RES     7,A         ;
FMS:  LD      (valor),A   ;

;      Fim do Modulo Sinal;

*****
*
*      Modulo de Envio de Dados P/ Gerador PWM
*
*****

;      Modulo PWM;
;      Entrada valor;

      LD      A,(valor)   ;
      OUT     (PORT_AD,A) ;

;      Fim do Modulo PWM;

*****
*
*      Modulo de Realocacao de Variaveis
*
*****

;      Modulo Realoc;
;      Entrada serro0, serro1, merro0,merro1;
;      Saida   serro1, merro1;

      LD      A,(serro0)  ;
      LD      (serro1),A  ;
      LD      A,(merro0)  ;
      LD      (merro1),A  ;

;
;
      RET

```

```

*****
*
*      Modulo de Multiplicacao
*
*****

```

```

;      Modulo Mult;
;      Entrada Reg.: D,E,A,B;
;      Saida   Reg.: HL,A;

MULT:   XOR      B          ;VERIFICACAO DE SINAIS
        LD       A,00H
        JP       Z,JUMP1
        CPL
JUMP1:  PUSH     AF
        LD       HL,00H
        SUB      A
        CP       E
        JP       Z,RETOR
        CP       D
        JP       Z,RETOR
        LD       B,08H
        LD       A,D
        LD       D,00H
MULT1:  ADD      HL,HL
        RLA
        JR       NC,CHCNT
        ADD      HL,DE
CHCNT:  DJNZ     MULT1
RETOR:  POP      AF
        RET

```

```

;      Fim do Modulo Mult
;

```

```

*****
*
*      Modulo de Soma
*
*****

```

```

;      Modulo Soma;
;      Entrada Reg.: HL,DE,A,B;
;      Saida   Reg.: HL,A;

SOMA:   XOR      B
        LD       A,B
        JP       Z,JUMP5
        CP       00H
        JP       NZ,JUMP6
        EX       DE,HL
JUMP6:  XOR      A

```

```

SBC      HL, DE
BIT      7, H
LD       A, 00H
RET
EX       DE, HL
LD       HL, 00H
XOR      A
SBC      HL, DE
LD       A, 0FFH
RET
JUMPS:   XOR      A
        ADC      HL, DE
        LD       A, B
        RET      NC
        LD       HL, 0FFFFH
        RET

```

```

;      Fim do Modulo Soma;

```

```

; MEMORIA RAM

```

	ORG	75FEH	
SINAL:	DS	1	; 75FEH - SINAL DO DADO DA AQUISICAO
MODULO:	DS	1	; 75FFH - MODULO DO DADO DA AQUISICAO
SREF:	DS	1	; 7600H - SINAL DA REFERENCIA
MREF:	DS	1	; 7601H - MODULO DA REFERENCIA
SQ0:	DS	1	; 7602H - SINAL DO COEFICIENTE Q0
MQ0:	DS	1	; 7603H - MODULO DO COEFICIENTE Q0
SQ1:	DS	1	; 7604H - SINAL DO COEFICIENTE Q1
MQ1:	DS	1	; 7605H - MODULO DO COEFICIENTE Q1
SQ2:	DS	1	; 7606H - SINAL DO COEFICIENTE Q2
MQ2:	DS	1	; 7607H - MODULO DO COEFICIENTE Q2
SERRO0:	DS	1	; 7608H - SINAL DO ERRO ECK)
MERRO0:	DS	1	; 7609H - MODULO DO ERRO ECK)
SERRO1:	DS	1	; 760AH - SINAL DO ERRO ECK-1)
MERRO1:	DS	1	; 760BH - MODULO DO ERRO ECK-1)
SERRO2:	DS	1	; 760CH - SINAL DO ERRO ECK-2)
MERRO2:	DS	1	; 760DH - MODULO DO ERRO ECK-2)
SU0:	DS	1	; 760EH - SINAL DA SAIDA UCK)
MU0:	DS	1	; 760FH - MODULO DA SAIDA UCK)
SSAIDA:	DS	1	; 7610H - SINAL DA SAIDA UCK+1)
MSAIDA:	DS	1	; 7611H - MODULO DA SAIDA UCK+1)
VALOR:	DS	1	; 7612H - VALOR DA SAIDA EM OFFSET-BINARIO
SAXC0:	DS	1	; 7613H - SINAL DA VARIABEL AUXILIAR AXC1
MAXC0:	DS	2	; 7614H - MODULO DA VARIABEL AUXILIAR AXC1
SAXC1:	DS	1	; 7616H - SINAL DA VARIABEL AUXILIAR AXC2
MAXC1:	DS	2	; 7617H - MODULO DA VARIABEL AUXILIAR AXC2
EW:	DS	1	; 7619H - HABILITACAO DE ESCRITA NA MEMORIA
PONT1:	DS	2	; 761AH - PONTEIRO DA TABELA DE POSICOES
PONT2:	DS	2	; 761CH - PONTEIRO DA TABELA DE FORCAS
EW1:	DS	1	; 761EH
END			

A P Ê N D I C E B

1- Objetivo

Este trabalho visa fazer a identificação dos parâmetros de motores de corrente contínua baseado no método proposto em [Lord, 1977], bem como simular o modelo obtido para comparação com os resultados experimentais.

2- Introdução teórica

Os motores CC têm larga utilização em sistemas de controle que necessitam de atuadores com boa performance de velocidade e precisão de movimentos. Dentre este tipo de motores, os mais utilizados atualmente, são os de ímã-permanente com relações torque-volume e torque-inércia elevadas.

Para determinação dos parâmetros de um motor cc com excitação separada (abreviaremos apenas por motor cc), existem diversos métodos. Alguns utilizam as técnicas de reposta em frequência, outros as técnicas do transiente de corrente em resposta a um degrau de tensão de armadura. Em [Lord 1977] é utilizada uma das técnicas de transiente, e também está proposta uma padronização da nomenclatura associada com motores cc.

Esta técnica divide os motores cc em oito modelos, quatro de primeira ordem e quatro de segunda ordem.

3- Procedimento Experimental

Como vimos anteriormente faz-se necessária a determinação da resposta de corrente a um degrau de tensão de armadura. Assim, torna-se necessário a utilização de um medidor adequado de corrente. Como não estava disponível um medidor direto, utilizamos um resistor "shunt" série para medir essa resposta de corrente.

Um diagrama do circuito utilizado neste experimento pode ser visto na figura 1.

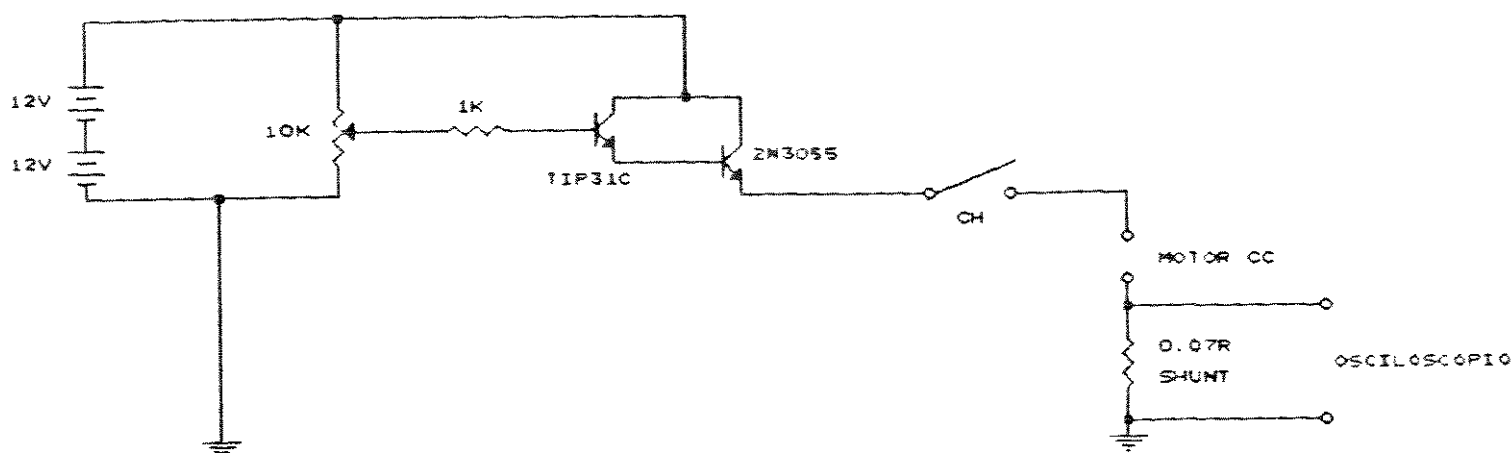


figura 1 - Diagrama do circuito utilizado nos experimentos.

A sequência de passos executados foram os seguintes :

- 1) Determinação do resistor "shunt" com auxílio de um ohmmetro de precisão;
- 2) Montagem do circuito de teste da figura 1;
- 3) Disparo de um degrau de tensão de armadura através do fechamento da chave;
- 4) Obtenção da curva de resposta da corrente obtida no osciloscópio via impressora;
- 5) Medida das correntes : $I_{(1)}$, $I_{(2)}$ e I_{ss} ;
- 6) Variação da tensão de armadura a partir de zero e verificar se o motor inicia a rotação imediatamente, se não medir a corrente com o qual ele começa a rodar;
- 7) Medida da velocidade final do motor utilizando uma luz estroboscópica;
- 8) Calculo dos valores através das fórmulas apresentadas em [Lord 1977];
- 9) Validação do modelo obtido via simulação através do programa motor.m no matlab.

4- Material Utilizado

1 - Equipamentos :

- Osciloscópio digital *Tectronics* modelo 2230;
- Impressora serial *Rima* modelo XT180;
- Medidor LCR *Escort* modelo ELC -130;
- Duas baterias de 12v *Good Light*.
- Luz estroboscópica *Frata*;

2 - Componentes :

- Associação de dois resistores de fio de $0.15 \Omega / 5 \text{ W}$, que medida no medidor LCR apresentou um valor de 0.07Ω ;
- Chave de duas posições.

3 - Motor (Dados do fabricante) :

- Motor cc imã-permanete *Lavill* :
 - 24 V
 - 3600 rpm
 - Saída com redutor ortogonal de 1:30
 - Corrente nominal : 2,5 A
 - Tacômetro digital : 100 pulsos/ 360°

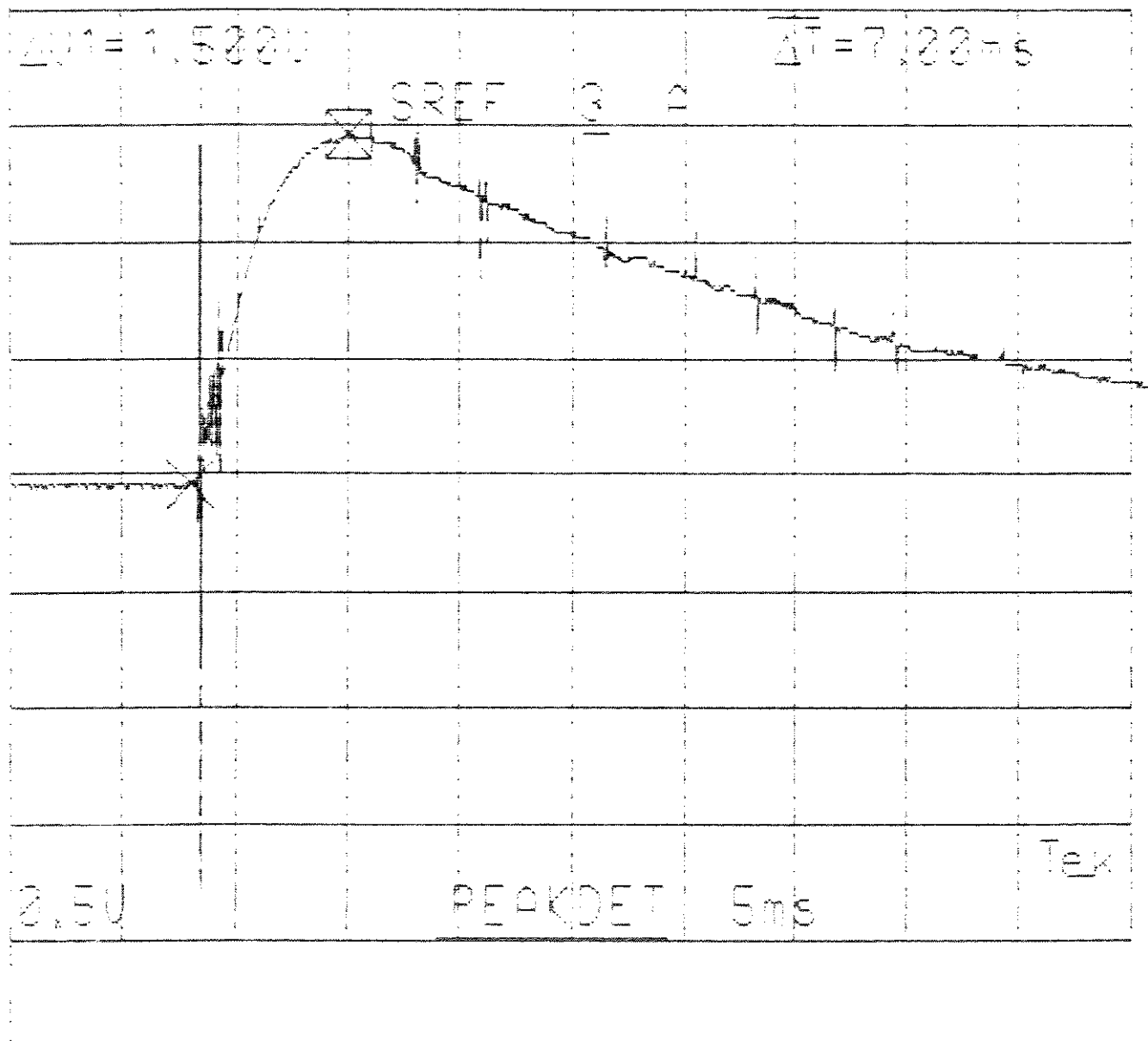
5- Resultados Experimentais

Os resultados experimentais da resposta da corrente de armadura a um degrau de tensão foram obtidas, indiretamente, através da medida de tensão no resistor "shunt" feita no osciloscópio e imprimidas via impressora serial. Esses resultados são apresentados nos gráficos 1 a 7.

Também foram feitas cinco medidas da corrente de zona morta $I_{(dz)}$, através da medida de tensão no resistor "shunt" $V_{(dz)}$. Estas estão apresentadas na tabela 1.

A velocidade final do motor medida foi : $\dot{W}_f = 3809 \text{ rpm}$.

TEKTRONIX 2230



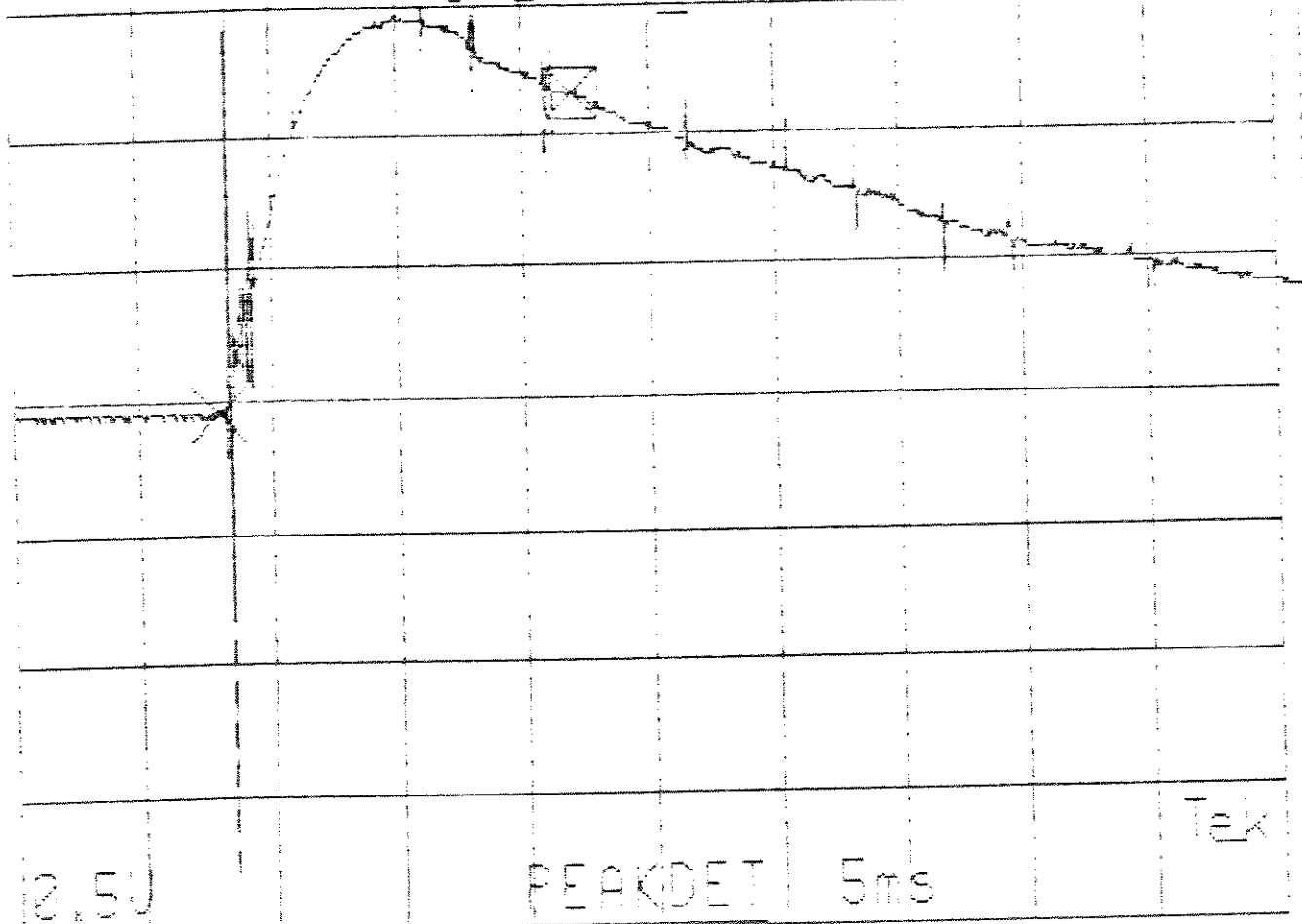
- Gráfico 1 - 1a. Medida da resposta de corrente com as marcações em t_1 . ($V_{shunt} \times s$)

TEKTRONIX 2232

$\Delta V_1 = 1.200V$

$\Delta t = 14.20 \mu s$

SREF 3 2



- Gráfico 2 - 1a. Medida da resposta de corrente com as marcações em 2.t1. (Vahunt x s)

TEKTRONIX 2232

$\Delta U = 1,800$

$\Delta T = 7,05 \cdot 10^{-5}$

SREF

3

2

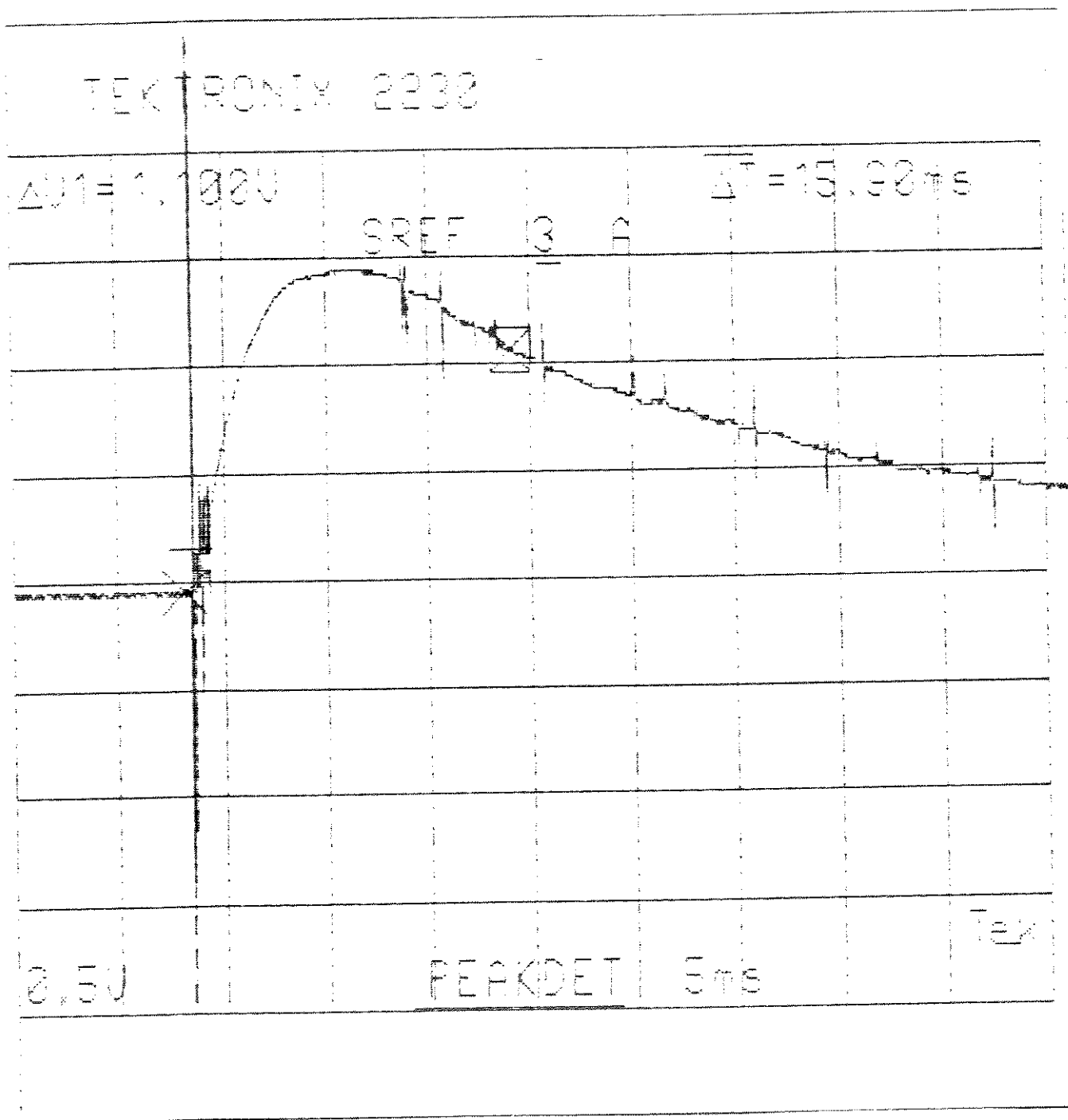
0.5V

PEAKDET

5ms

Tek

- Gráfico 3 - 2a. Medida da resposta de corrente com as marcações em t_1 . (Vahunt x s)



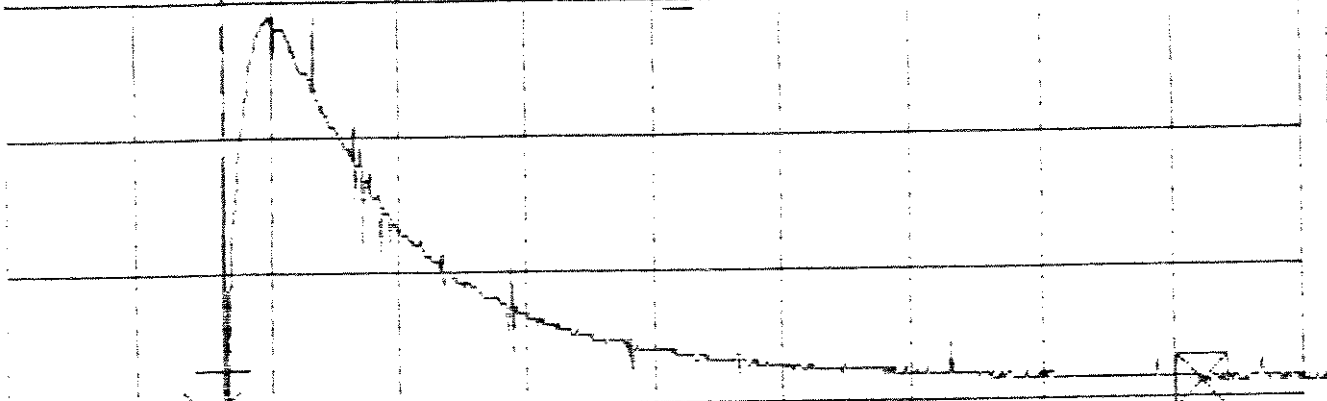
- Gráfico 4 - 2a. Medida da resposta de corrente com as marcações em 2.t_s. (V_{ahunt} x s)

TEKTRONIX 2230

$\Delta V_1 = 0.100V$

SREF 3 2

$\Delta T = 153.4ms$



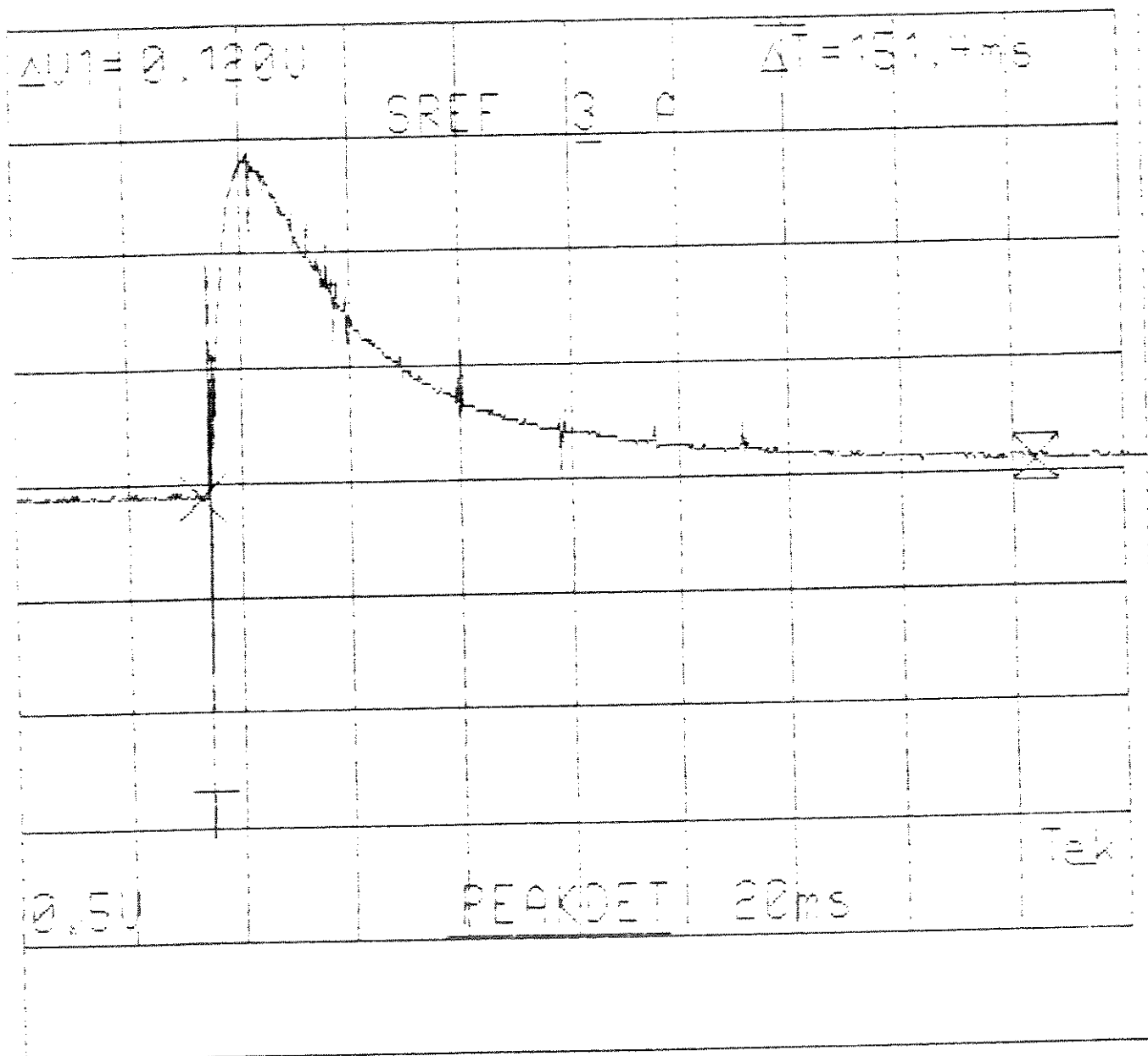
0.5V

PEAKDET 20ms

Tek

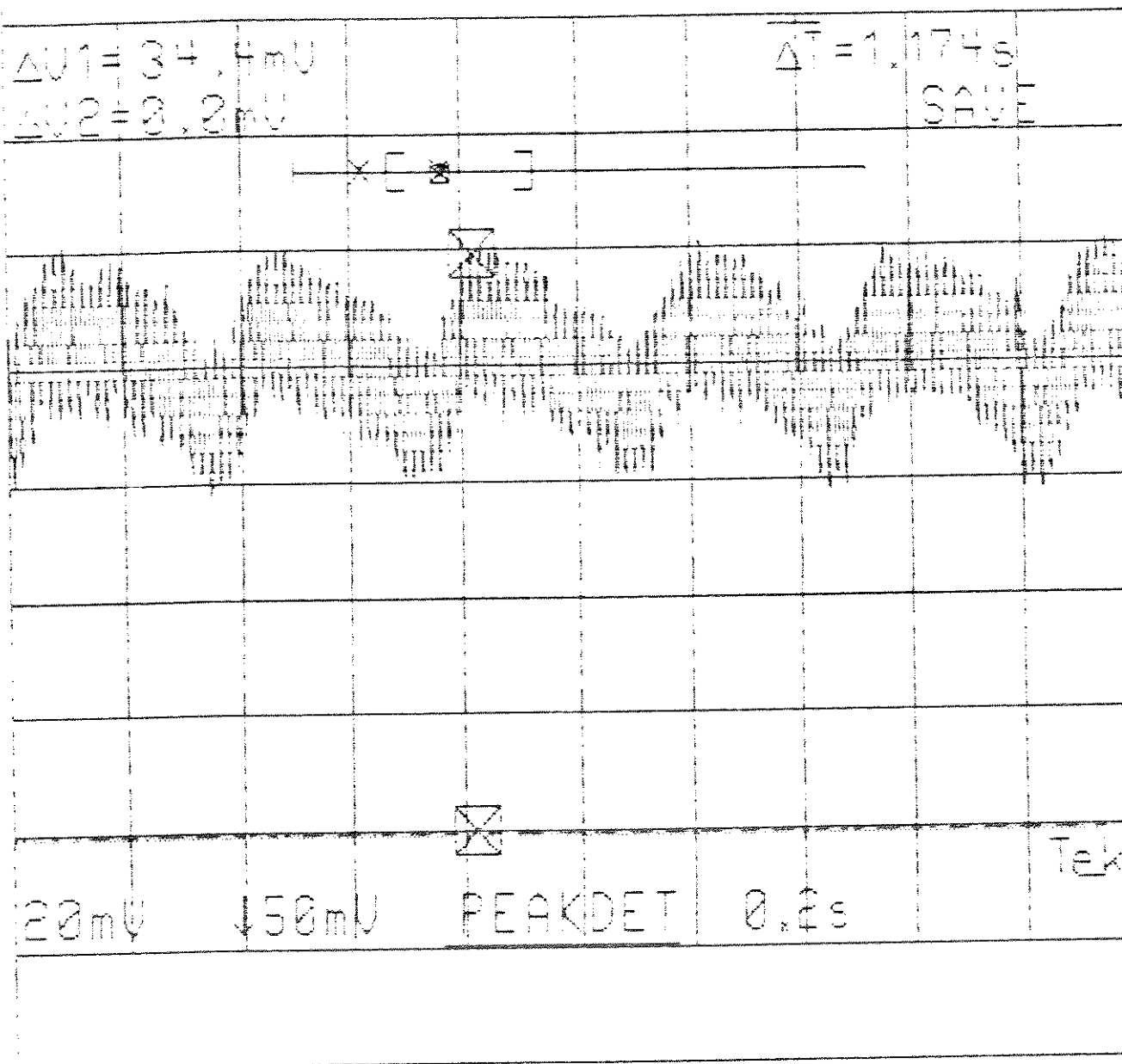
- Gráfico 5 - 1a. Medida da resposta da corrente com a base de tempo estendida para observação da corrente estacionária. (Vahunt x s)

TEKTRONIX 2232



- Gráfico 6 - 2a. Medida da resposta da corrente com a base de tempo estendida para observação da corrente estacionária. (Vahunt x s)

TEKTRONIX 2232



- Gráfico 7 - Sinal do ruído da corrente estacionária.
(Vshunt x s)

Medida	V(tdz) (mV)
1	86
2	80
3	90
4	76
5	74

Tabela 1 - Tensão no resistor "shunt" na zona morta.

Utilizando esses dados obtidos e substituindo-os nas equações dadas em [Lord 1977], teremos os seguintes parâmetros:

- 1a. Medida :

- Constante de tempo elétrica : $\tau_e = 2,8 \text{ ms}$
- Constante de tempo mecânica : $\tau_m = 21 \text{ ms}$
- Constante de tempo de amortecimento : $\tau_b = 2,079 \text{ s}$
- Constante de voltagem : $K_e = 0,057$
- Constante de torque : $K_t = K_e = 0,057$
- Constante de torque de atrito : $T_f = 65,78 \text{ mN.m}$
- Indutância de armadura : $L_a = 2,5 \text{ mH}$
- Resistência de armadura : $R_a = 0,896 \Omega$
- Coeficiente de atrito viscoso : $B = 36,63 \times 10^{-6} \text{ Kg.m}^2/\text{s}$
- Coeficiente de inércia : $J = 76,15 \times 10^{-6} \text{ Kg.m}^2$
- Tempo associado a zona morta : $t_{dz} = 123,07 \mu\text{s}$

- 2a. Medida :

- Constante de tempo elétrica : $\tau_e = 3,6 \text{ ms}$
- Constante de tempo mecânica : $\tau_m = 18 \text{ ms}$
- Constante de tempo de amortecimento : $\tau_b = 0,885 \text{ s}$
- Constante de voltagem : $K_e = 0,057$
- Constante de torque : $K_t = K_e = 0,057$
- Constante de torque de atrito : $T_f = 65,78 \text{ mN.m}$
- Indutância de armadura : $L_a = 3,05 \text{ mH}$
- Resistência de armadura : $R_a = 0,844 \Omega$

- Coeficiente de atrito viscoso : $B = 78,56 \times 10^{-6} \text{ Kg.m}^2/\text{s}$
- Coeficiente de inércia : $J = 69,56 \times 10^{-6} \text{ Kg.m}^2$
- Tempo associado a zona morta : $t_{dz} = 151,3 \text{ } \mu\text{s}$

6- Análise dos Dados Obtidos

Como a priori estávamos querendo determinar também a resistência de armadura do motor, e portanto, não conhecíamos o seu valor, não podíamos determinar uma resistência adequada para o "shunt". Assim, optamos pela utilização da menor resistência mensurável pelo medidor LCR. Após a determinação do valor de R_a através dos experimentos, verificamos que o valor do "shunt" utilizado pode ser considerado adequado, pois seu valor é dez vezes menor do que R_a .

Para uma análise dos resultados obtidos devemos verificar se o modelo obtido aproxima-se das características reais do motor. Assim torna-se importante a utilização de uma ferramenta que nos auxilie na simulação o modelo do motor cc.

Assim, optamos por usar o programa *Matlab* para fazer essa simulação. O diagrama de blocos do modelo simulado está representado na figura 2.

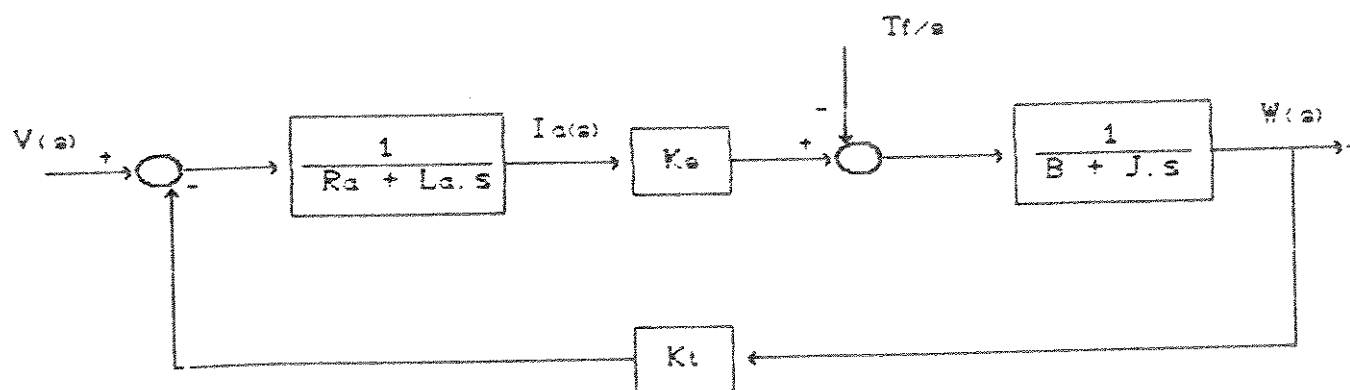


Figura 2 - Diagrama de blocos utilizado para modelar o motor.

A relação entre $W(s)$ e $V(s)$ é dada por :

$$W(s) = \frac{K_t \cdot V(s) \cdot s - (R_a + L_a \cdot s) \cdot T_f}{[(J \cdot s + B) \cdot (R_a + L_a \cdot s) + K_t^2] \cdot s} \quad (1)$$

E a relação entre $I_a(s)$ e $V(s)$ é dada por :

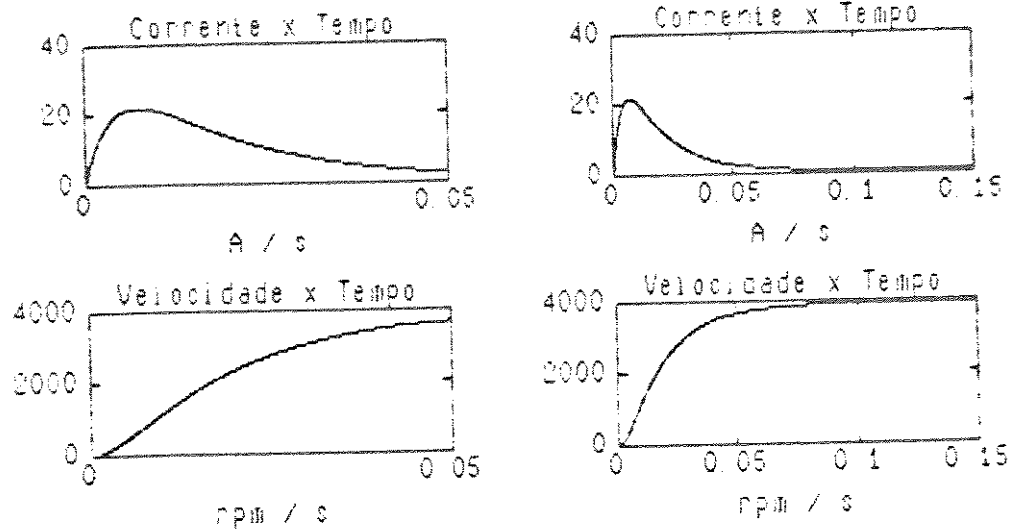
$$I_a(s) = \frac{A \cdot s^3 + B \cdot s^2 + C \cdot s + D}{E \cdot s^4 + F \cdot s^3 + G \cdot s^2 + H \cdot s} \quad (2)$$

onde :

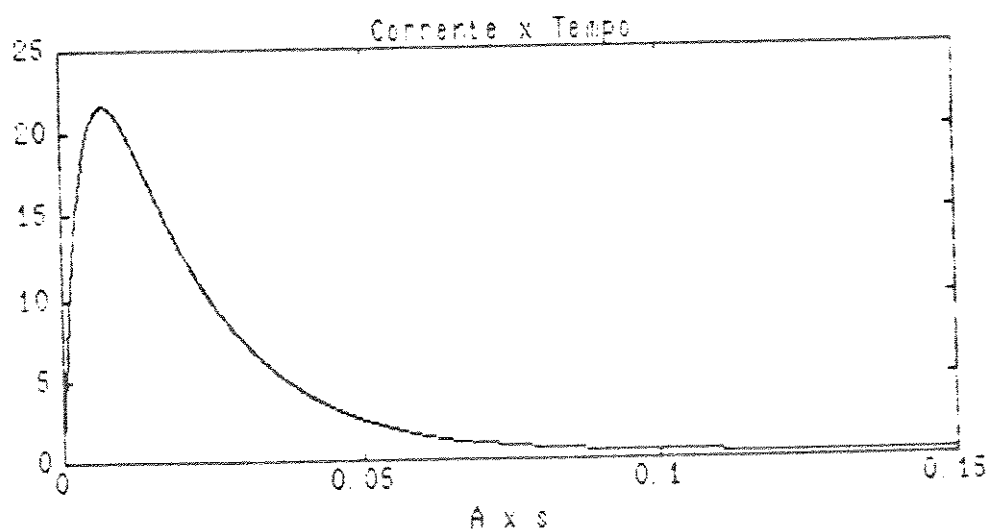
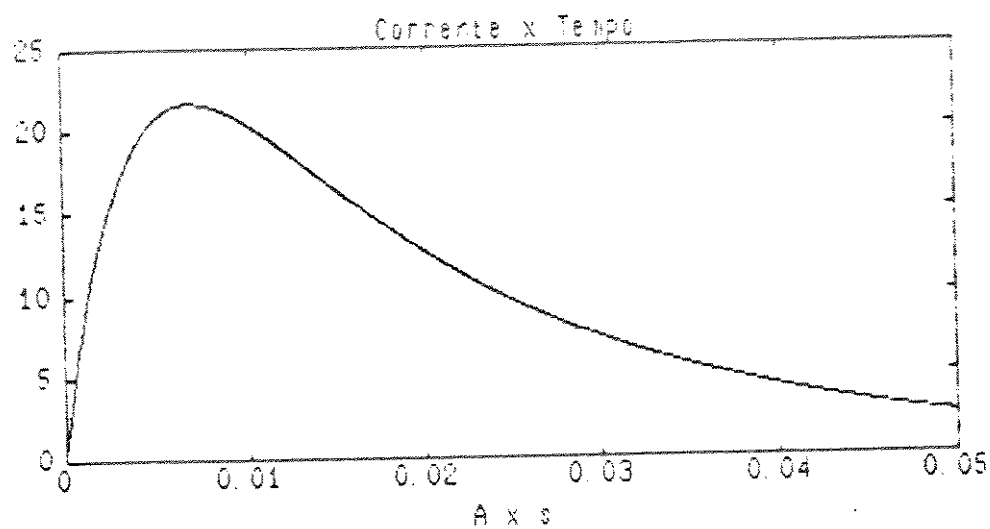
$$\begin{aligned} A &= V(s) \cdot J \cdot L_a ; \\ B &= V(s) \cdot (J \cdot R_a + B \cdot L_a) ; \\ C &= V(s) \cdot B \cdot R_a + K_t \cdot T_f \cdot L_a ; \\ D &= K_t \cdot T_f \cdot R_a ; \\ E &= J \cdot L_a^2 ; \\ F &= 2 \cdot J \cdot L_a \cdot R_a + B \cdot L_a^2 ; \\ G &= J \cdot R_a^2 + 2 \cdot B \cdot R_a \cdot L_a + K_t^2 \cdot L_a ; \\ H &= B \cdot R_a^2 + K_t^2 \cdot R_a ; \end{aligned}$$

Para facilitar essa simulação construímos um programa dentro do *Matlab* chamado *motor.m*, listado no anexo A, e obtivemos os seguintes gráficos 8 e 9 (que são aumentados em escala nos gráficos 8a, 8b, 9a e 9b).

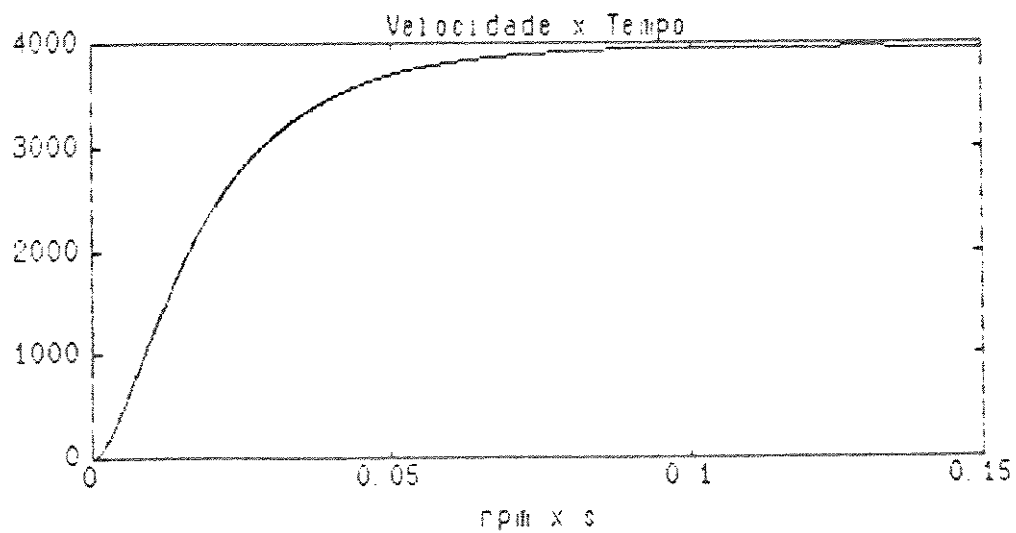
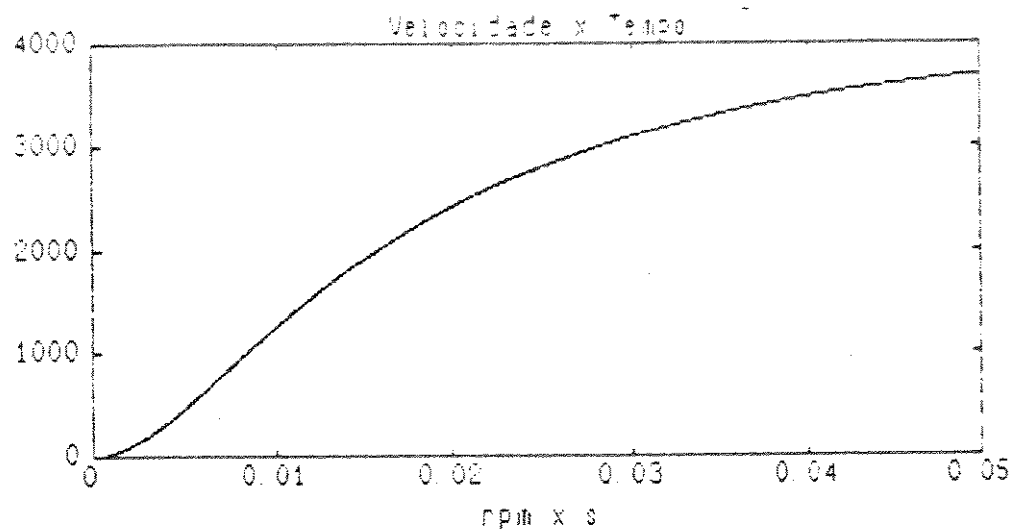
Apesar de termos feito apenas duas medições, podemos fazer uma breve análise baseada nos valores dos parâmetros obtidos. Constatamos que os parâmetros da parte elétrica do modelo não se alteraram significativamente. Já os parâmetros da parte mecânica, principalmente o coeficiente de atrito viscoso, variaram de forma significativa. Ao que tudo indica isto ocorreu porque o motor utilizado possui um sistema de redução de velocidade, e constatamos, com o auxílio da luz estroboscópica, que este forçava variações na velocidade do eixo do motor. Isto também pode ser constatado no gráfico experimental 7, onde nota-se que aproximadamente a cada 400ms o motor fica sujeito a um



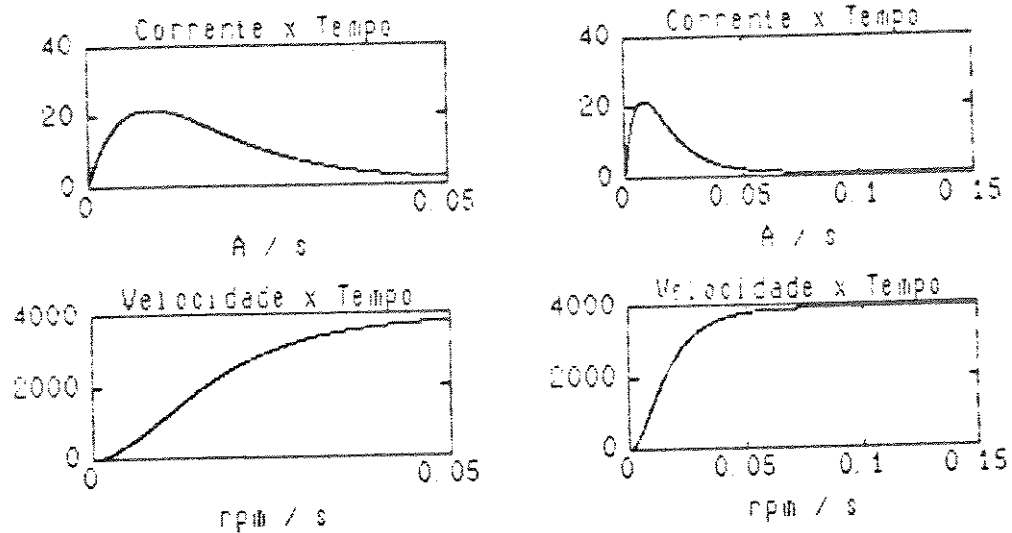
- Gráfico 8 - Simulação das curvas características de corrente de armadura e velocidade do motor cc em resposta a um degrau de tensão de armadura utilizando os parâmetros obtidos na 1ª. medida.



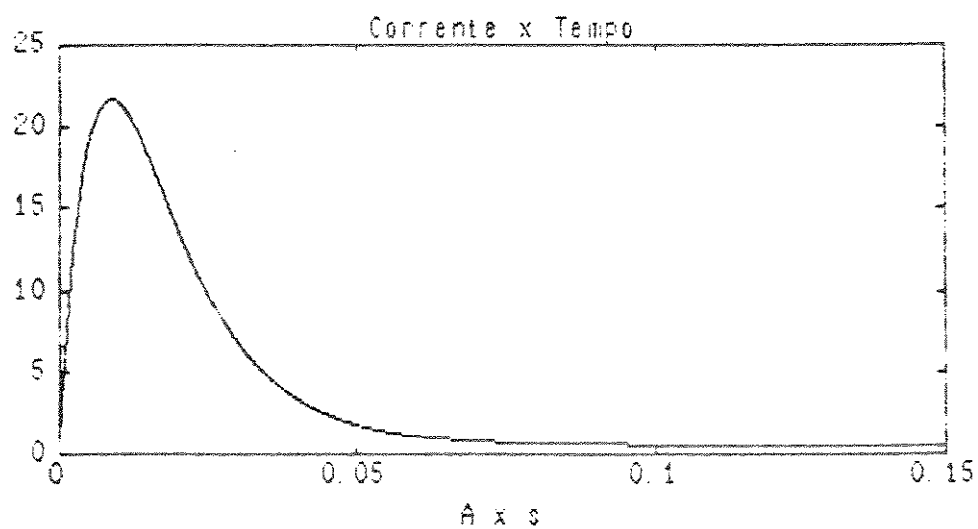
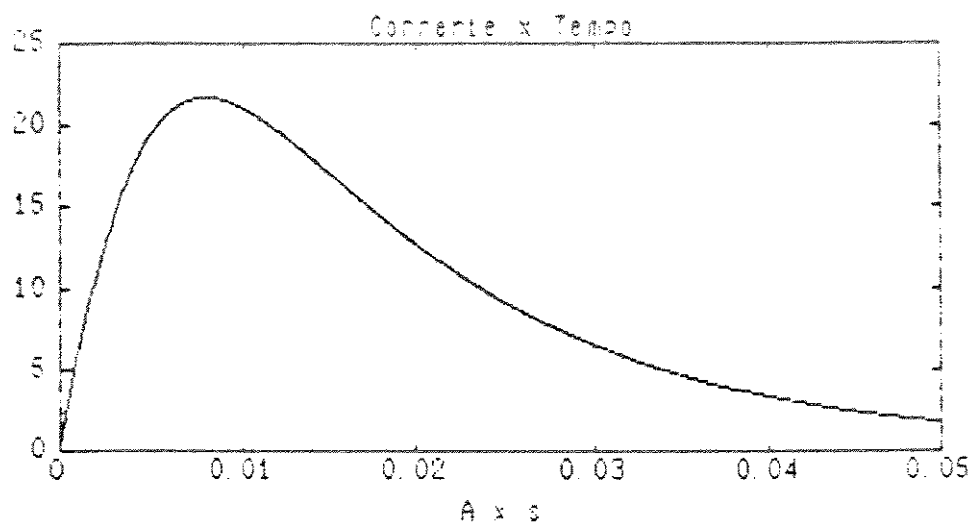
- Gráfico 8a. - Simulação da resposta de corrente utilizando os parâmetros obtidos na 1ª. medida.



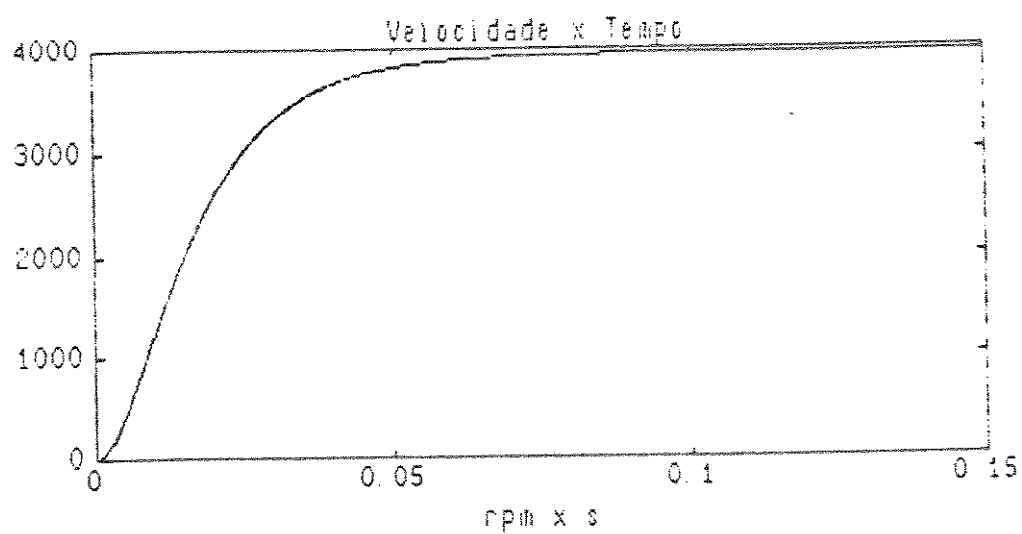
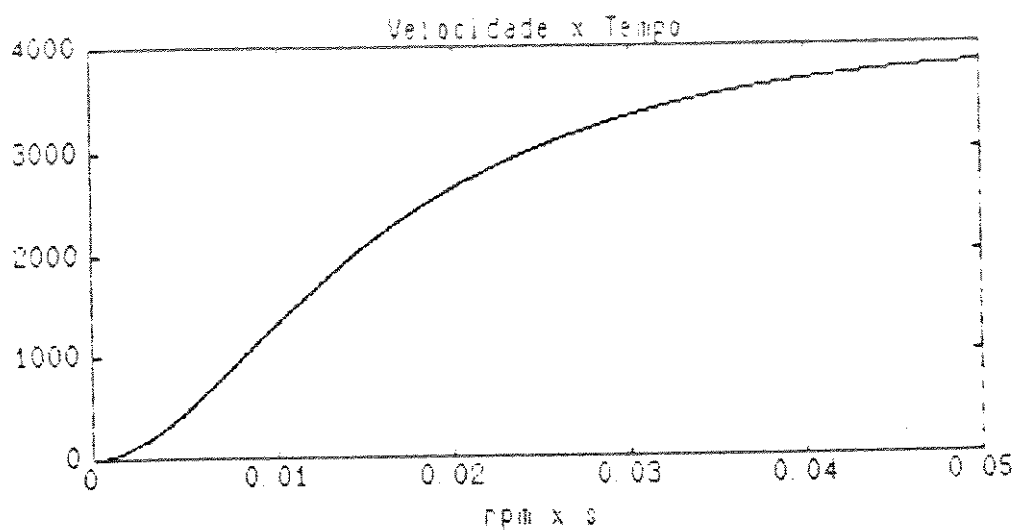
- Gráfico 8b. - Simulação da resposta de velocidade utilizando os parâmetros obtidos na 1a. medida.



- Gráfico 9 - Simulação das curvas características de corrente de armadura e velocidade do motor cc em resposta a um degrau de tensão de armadura utilizando os parâmetros obtidos na 2a. medida.



- Gráfico 9a. - Simulação da resposta da corrente utilizando os parâmetros obtidos na 2a. medida.



- Gráfico 9b. - Simulação da resposta de velocidade utilizando os parâmetros obtidos na 2a. medida.

tempo de maior carga e a um tempo de menor carga provocado por irregularidades do sistema de redução de velocidade.

Através dos gráficos experimentais (1 a 6), observamos que o ruído provocado pelas escovas do motor é de grande intensidade e frequência. Isso pode impor dificuldades na aquisição de valores corretos da corrente, principalmente se esta estiver sendo feita digitalmente.

7- Conclusão

Após verificarmos atentamente os gráficos obtidos experimentalmente (gráficos 1 a 7), e os gráficos obtidos via simulação através do modelo baseado nas equações (1) e (2), podemos concluir que os valores dos parâmetros obtidos experimentalmente são próximos dos valores reais, pois os gráficos apresentam grande semelhança. Logo, o método proposto por [Lord 1977] é muito eficiente para determinação dos parâmetros de um motor cc com excitação independente.

A N E X O A

Rotina principal (MOTOR.M)

Simulacao da Resposta de Motor CC ao degrau

Todos os parametros devem estar no sistema SI

```
clc
clear

t1= 50e-3;
t2= 150e-3;

while 1
    cla
    esc= ['EP'
          'AP'
          'SI'];

    help menu1
    n= input('Escolha a alternativa : ');
    if(n == 0)
        cla
        break
    end
    esc= esc(n,:);
    eval(esc)
end
```

Subrotina que apresenta o menu principal (MFNU1.M)

SIMULACAO DA RESPOSTA DE CORRENTE A ENTRADA DEGRAU DE MOTOR CC

- 1) Entrada de Parametros
- 2) Alteracao de Parametros
- 3) Simular
- 0) Saida

Subrotina que apresenta menu de alteracao de parametros (MENU2.M)

Alteracao de Parametros

- 1) Degrau de tensao
- 2) Inercia
- 3) Atrito viscoso
- 4) Resistencia de armadura
- 5) Indutancia de armadura
- 6) Torque de atrito seco
- 7) Constante de tempo do motor

- 0) Retorno a menu principal

Subrotina para entrada de parametros (EP.M)

cla

Entrada de parametros

```
U= input('Qual o degrau de tensao de armadura (U)? ');
J= input('Qual a constante de inercia (J)? ');
B= input('Qual a constante de atrito viscoso (B)? ');
R= input('Qual a resistencia de armadura (Ra)? ');
L= input('Qual a indutancia de armadura (La)? ');
Tf= input('Qual o torque de atrito seco (Tf)? ');
Kt= input('Qual a constante de torque do motor (Kt)? ');
```

Subrotina para alteracao de parametros (AP.M)

Alteracao de parametros

```
while 1
    cla
    help menu2
    ea= input('Escolha a alternativa : ');
    if(ea == 0)
        break
    end
    if(ea == 1)
        cla
        V= input('Qual o degrau de tensao de armadura (V)? ');
    end
    if(ea == 2)
        cla
        J= input('Qual a constante de inercia (J)? ');
    end
    if(ea == 3)
        cla
        B= input('Qual a constante de atrito viscoso (B)? ');
    end
    if(ea == 4)
        cla
        R= input('Qual a resistencia de armadura (Ra)? ');
    end
    if(ea == 5)
        cla
        L= input('Qual a indutancia de armadura (La)? ');
    end
    if(ea == 6)
        cla
        Tf= input('Qual o torque de atrito seco (Tf)? ');
    end
    if(ea == 7)
        cla
        Kt= input('Qual a constante de torque do motor (Kt)? ');
    end
end
end
```

Subrotina para simulacao e geracao de graficos (SI.M)

```
cla
disp('Os Tempos para os graficos sao 50ms e 150ms');
alt= 'alt';
s= 's';
alt= input('Voce deseja altera-los? (s/n) : ');

if(alt == s)
    cla
    t1= input('Qual o tempo de resposta desejado para curva 1 (s)? ');
    t2= input('Qual o tempo de resposta desejado para curva 2 (s)? ');
end

cla
tp1= t1/1000;
T1= 0:tp1:t1;
tp2= t2/1000;
T2= 0:tp2:t2;

inum= [0 V*J*L V*J*R+V*B*L V*B*R+Kt*Tf*L Kt*Tf*R];
iden= [J*L*L 2*J*L*R+B*L*L J*R*R+2*B*R*L+Kt*Kt*L B*R*R+Kt*Kt*R 0];

wnum= [0 0 Kt*V-L*Tf -R*Tf];
wden= [J*L J*R+B*L B*R+Kt*Kt 0];

disp('Aguarde Estou Calculando ia(t) !!!');
I1= step(inum,iden,T1);
I2= step(inum,iden,T2);

disp('');
disp('Aguarde Estou Calculando w(t) !!!');
W1= step(wnum,wden,T1);
W2= step(wnum,wden,T2);
W1r= W1*(60/(2*pi));
W2r= W2*(60/(2*pi));

vai='vai';
vai=input('Voce deseja graficos separados (S/N)? : ');

if(vai==s)
    cla
    plot(T1,I1);
    title('Corrente x Tempo');
    xlabel(' A x s ');
    pause

    cla
    plot(T2,I2);
    title('Corrente x Tempo');
    xlabel(' A x s ');
    pause
```

```

cla
plot(T1,W1r);
title('Velocidade x Tempo');
xlabel('rpm x s');
pause

```

```

cla
plot(T2,W2r);
title('Velocidade x Tempo');
xlabel('rpm x s');
pause

```

end

```

cla
subplot(221),plot(T1,I1);
title('Corrente x Tempo');
xlabel('A / s');
subplot(222),plot(T2,I2);
title('Corrente x Tempo');
xlabel('A / s');
subplot(223),plot(T1,W1r);
title('Velocidade x Tempo');
xlabel('rpm / s');
subplot(224),plot(T2,W2r);
title('Velocidade x Tempo');
xlabel('rpm / s');
pause
clf

```

A P Ê N D I C E C

Model File: TESES.TUT
 Date: 7 / 31 / 1991
 Time: 20 : 58
 Timing: 100.000E-06 ,DELTA ; 1.0000 ,RANGE
 PlotBlocks and Scales:
 Format:
 BlockNo, Plot-MINimum, Plot-MAXimum; Comment
 Horz: 0 , 0.0000 , 0.3750000 ; Time
 Y1: 165 , 0.0000 , 0.0010000 ;
 Y2: , , ;
 Y3: , , ;
 Y4: , , ;

24.0000	10	CON			
0.5740000	15	GAI	10		
0.0019600	20	ATT	15	-25	-65
0.1649600	25	GAI	30		
0.0000	30	INT	20		
2.0000	35	ATT	30		
	40	SUM	35	-75	
2.0000	45	ATT	65		
0.0000	50	INT	40		
99.997E+03	55	GAI	50		
9.0000	60	GAI	40		
	65	SUM	55	60	
2.473E+03	66	ATT	140	145	
0.0020000	70	ATT	65	-80	-66
0.0000	75	INT	70		
0.0050000	80	GAI	75		
	85	ABS	70		
2.473E+03	90	GAI	150		
	95	SUM	85	-100	
	100	ABS	90		
	105	IFE	95	75	110
0.0000	110	CON			
2.473E+03	120	ATT	105		
1.000E-06	125	ADL	120	-155	
0.0000	130	INT	120		
0.0000	135	INT	125		
10.000E+03	140	GAI	135		
5.0000	145	GAI	125		
0.3000000	150	ATT	140	145	-175
			-180		
0.0000	155	INT	150		
	160	SUM	155	-190	
0.0010000	161	CON			
	162	SUM	161	-165	
	163	IFE	162	110	160

0.0000	165 INT	155		
0.0000	170 INT	163		
99.997E+03	175 GAI	170		
8.0000	180 GAI	163		
0.5000000	185 ATT	175	180	-205
		-235		
0.0000	190 INT	185		
0.0000	195 INT	190		
	200 SUM	195	-225	
0.0010000	201 CON			
	202 SUM	201	-195	
	203 IFE	202	110	195
99.997E+03	205 GAI	203		
8.0000	210 ATT	205		
	220 SUM	232	-210	
0.0000	225 INT	220		
	230 SUM	232	-220	
	231 IFE	202	110	230
	232 IFE	202	110	190
8.0000	235 GAI	231		